```
# Python source code for
# Generated by Cython 0.29.21
# Cython: C++ language_level=3
# Auto-pickling is disabled on all types.
# distutils: sources = /home/runner/work/_temp/tmpqqqqqqqq.pyx
# distutils: extra_compile_args = -O3 -ffast-math -fopenmp -fopenmp-simd -mavx2 -mfma -mf16c -Ofast -ffast-math -Wno-unreachable-code -Wno-unreachable-code-fallthrough -Wno-unreachable-code-return -Wno-unreachable-code-loop -Wno-maybe-uninitialized -Wno-ignored-attributes -Wno-cpp -Wno-deprecated-declarations -Wno-parentheses -Wno-pragma-once -Wno-unknown-pragmas -Wno-unknown-warning-option -Wno-unknown-warning -Wno-ignored-pragmas -Wno-ignored-attributes -Wno-ignored-qualifiers -Wno-ignored-pragmas
```

```
  update the black and yellow checkered flag
  FUNCTION UPDATECHECKERED(CHECKERED, DT)
  update the black and white checkered flag
  FUNCTION UPDATEWHITECHECKERED(WHITECHECKERED, DT)
  update the red and white checkered flag
  FUNCTION UPDATEREDWHITECHECKERED(REDWHITECHECKERED, DT)
  update the red and yellow checkered flag
  FUNCTION UPDATEREDYELLOWCHECKERED(REDYELLOWCHECKERED, DT)
  update the yellow and black checkered flag
  FUNCTION UPDATEYELLOWBLACKCHECKERED(YELLOWBLACKCHECKERED, DT)
  update the yellow and white checkered flag
  FUNCTION UPDATEYELLOWWHITECHECKERED(YELLOWWHITECHECKERED, DT)
  update the yellow and red checkered flag
  FUNCTION UPDATEYELLOWREDCHECKERED(YELLOWREDCHECKERED, DT)
  update the white and black checkered flag
  FUNCTION UPDATEWHITEBLACKCHECKERED(WHITEBLACKCHECKERED, DT)
  update the white and red checkered flag
  FUNCTION UPDATEWHITEREDCHECKERED(WHITEREDCHECKERED, DT)
  update the black and red checkered flag
  FUNCTION UPDATEBLACKREDCHECKERED(BLACKREDCHECKERED, DT)
  update the black and yellow checkered flag
  FUNCTION UPDATEBLACKYELLOWCHECKERED(BLACKYELLOWCHECKERED, DT)
  update the black and white checkered flag
  FUNCTION UPDATEBLACKWHITECHECKERED(BLACKWHITECHECKERED, DT)
  update the red and black checkered flag
  FUNCTION UPDATEREDBLACKCHECKERED(REDBLACKCHECKERED, DT)
  update the red and white checkered flag
  FUNCTION UPDATEREDWHITECHECKERED(REDWHITECHECKERED, DT)
  update the red and yellow checkered flag
  FUNCTION UPDATEREDYELLOWCHECKERED(REDYELLOWCHECKERED, DT)
  update the white and black checkered flag
  FUNCTION UPDATEWHITEBLACKCHECKERED(WHITEBLACKCHECKERED, DT)
  update the white and red checkered flag
  FUNCTION UPDATEWHITEREDCHECKERED(WHITEREDCHECKERED, DT)
  update the white and yellow checkered flag
  FUNCTION UPDATEWHITEYELLOWCHECKERED(WHITEYELLOWCHECKERED, DT)
  update the yellow and black checkered flag
  FUNCTION UPDATEYELLOWBLACKCHECKERED(YELLOWBLACKCHECKERED, DT)
  update the yellow and white checkered flag
  FUNCTION UPDATEYELLOWWHITECHECKERED(YELLOWWHITECHECKERED, DT)
  update the yellow and red checkered flag
  FUNCTION UPDATEYELLOWREDCHECKERED(YELLOWREDCHECKERED, DT)
  update the black and white
  FUNCTION UPDATEBLACKWHITE(BLACKWHITE, DT)
  update the black and red
  FUNCTION UPDATEBLACKRED(BLACKRED, DT)
  update the black and yellow
  FUNCTION UPDATEBLACKYELLOW(BLACKYELLOW, DT)
  update the white and black
  FUNCTION UPDATEWHITEBLACK(WHITEBLACK, DT)
  update the white and red
  FUNCTION UPDATEWHITERED(WHITERED, DT)
  update the white and yellow
  FUNCTION UPDATEWHITEYELLOW(WHITEYELLOW, DT)
  update the yellow and black
  FUNCTION UPDATEYELLOWBLACK(YELLOWBLACK, DT)
  update the yellow and white
  FUNCTION UPDATEYELLOWWHITE(YELLOWWHITE, DT)
  update the yellow and red
  FUNCTION UPDATEYELLOWRED(YELLOWRED, DT)
  update the red and black
  FUNCTION UPDATEREDBLACK(REDBLACK, DT)
  update the red and white
  FUNCTION UPDATEREDWHITE(REDWHITE, DT)
  update the red and yellow
  FUNCTION UPDATEREDYELLOW(REDYELLOW, DT)
  update the black
  FUNCTION UPDATEBLACK(BLACK, DT)
  update the white
  FUNCTION UPDATEWHITE(WHITE, DT)
  update the yellow
  FUNCTION UPDATEYELLOW(YELLOW, DT)
  update the red
  FUNCTION UPDATERED(R
```

```
<?xml version="1.0" encoding="UTF-8" ?>
<rss version="2.0" xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:sy="http://purl.org/rss/1.0/modules/syndication/" xmlns:slash="http://purl.org/rss/1.0/modules/slash/" xmlns:wfw="http://wellformedweb.org/CommentAPI/" xmlns:media="http://search.yahoo.com/mrss/" xmlns:atom10="http://www.w3.org/2005/Atom" xmlns:opensearch="http://a9.com/-/spec/opensearchrss/1.0/" xmlns:geo="http://www.w3.org/2003/01/geo/wgs84_pos#" xmlns:georss="http://www.georss.org/georss" xmlns:gml="http://www.opengis.net/gml" xmlns:thr="http://purl.org/rss/1.0/modules/threading/" xmlns:feedburner="http://rssnamespace.org/feedburner/ext/1.0" xmlns:feedburner:origlink="http://rssnamespace.org/feedburner/extensions/1.0" xmlns:media:rating="http://search.yahoo.com/mrss/rating" xmlns:media:copyright="http://search.yahoo.com/mrss/copyright" xmlns:media:text="http://search.yahoo.com/mrss/text" xmlns:media:hash="http://search.yahoo.com/mrss/hash" xmlns:media:player="http://search.yahoo.com/mrss/player" xmlns:media:credit="http://search.yahoo.com/mrss/credit" xmlns:media:category="http://search.yahoo.com/mrss/category" xmlns:media:rss="http://search.yahoo.com/mrss/" xmlns:media:thumbnail="http://search.yahoo.com/mrss/thumbnail" xmlns:media:restriction="http://search.yahoo.com/mrss/restriction" xmlns:media:adultcontent="http://search.yahoo.com/mrss/adultcontent" xmlns:media:keywords="http://search.yahoo.com/mrss/keywords" xmlns:media:scene="http://search.yahoo.com/mrss/scene" xmlns:media:comments="http://search.yahoo.com/mrss/comments" xmlns:media:license="http://search.yahoo.com/mrss/license" xmlns:media:peerlink="http://search.yahoo.com/mrss/peerlink" xmlns:media:location="http://search.yahoo.com/mrss/location" xmlns:media:poll="http://search.yahoo.com/mrss/poll" xmlns:media:uploaded="http://search.yahoo.com/mrss/uploaded" xmlns:media:uploadedby="http://search.yahoo.com/mrss/uploadedby" xmlns:media:expression="http://search.yahoo.com/mrss/expression" xmlns:media:price="http://search.yahoo.com/mrss/price" xmlns:media:subTitle="http://search.yahoo.com/mrss/subTitle" xmlns:media:pubStart="http://search.yahoo.com/mrss/pubStart" xmlns:media:pubStop="http://search.yahoo.com/mrss/pubStop" xmlns:media:priority="http://search.yahoo.com/mrss/priority" xmlns:media:rights="http://search.yahoo.com/mrss/rights" xmlns:media:subCategory="http://search.yahoo.com/mrss/subCategory" xmlns:media:peerLinkFollow="http://search.yahoo.com/mrss/peerLinkFollow" xmlns:media:responses="http://search.yahoo.com/mrss/responses" xmlns:media:publication="http://search.yahoo.com/mrss/publication" xmlns:media:addtime="http://search.yahoo.com/mrss/addtime" xmlns:media:groups="http://search.yahoo.com/mrss/groups" xmlns:media:statusbar="http://search.yahoo.com/mrss/statusbar" xmlns:media:vdisplay="http://search.yahoo.com/mrss/vdisplay" xmlns:media:vstyle="http://search.yahoo.com/mrss/vstyle" xmlns:media:adBlock="http://search.yahoo.com/mrss/adBlock" xmlns:media:status="http://search.yahoo.com/mrss/status" xmlns:media:mature="http://search.yahoo.com/mrss/mature" xmlns:media:country="http://search.yahoo.com/mrss/country" xmlns:media:language="http://search.yahoo.com/mrss/language" xmlns:media:director="http://search.yahoo.com/mrss/director" xmlns:media:starring="http://search.yahoo.com/mrss/starring" xmlns:media:publication_info="http://search.yahoo.com/mrss/publication_info" xmlns:media:production_info="http://search.yahoo.com/mrss/production_info" xmlns:media:musicinfo="http://search.yahoo.com/mrss/musicinfo" xmlns:media:artist="http://search.yahoo.com/mrss/artist" xmlns:media:recording_info="http://search.yahoo.com/mrss/recording_info" xmlns:media:lyrics="http://search.yahoo.com/mrss/lyrics" xmlns:media:product="http://search.yahoo.com/mrss/product" xmlns:media:publication_info="http://search.yahoo.com/mrss/publication_info" xmlns:media:rental="http://search.yahoo.com/mrss/rental" xmlns:media:purchase="http://search.yahoo.com/mrss/purchase" xmlns:media:publication="http://search.yahoo.com/mrss/publication
```

```
<div_class="header_wrapper">
<div_class="header_inner_wrapper">
<div_class="header_left_wrapper">
<div_class="header_logo_wrapper">
<div_class="header_logo_inner_wrapper">
<div_class="header_logo_text_wrapper">
<div_class="header_logo_text_inner_wrapper">
<div_class="header_logo_text_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
<div_class="header_logo_text_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_inner_wrapper">
```

```
// Author of the Shear Force Diagram !!
// Author of the Bending Moment !!
// Author of the Deflection !!
// Author of the Slope !!

// Author of the Shear Force Diagram !!
// Author of the Bending Moment !!

// The Shear Force Diagram Function !!
function drawSFD(){
    // Find the left and right edges of the beam
    var leftEdgeX = Math.min(beamStartX, beamStartX+beamLength);
    var rightEdgeX = Math.max(beamStartX, beamStartX+beamLength);

    // Find the left and right edge of the beam
    var leftEdgeY = getBeamYatX(leftEdgeX);
    var rightEdgeY = getBeamYatX(rightEdgeX);
    if (leftEdgeX == rightEdgeX){
        leftEdgeY = beamStartY;
        rightEdgeY = beamStartY+beamLength*Math.sin(beamAngle);
        if(beamAngle==0.0&&beamLength==0.0){leftEdgeY = mouseY;}
    }

    // Replace the point and the high force value !!
    var highestForceValue = 0.0;
    var highestForceIndex = 0;
    for(var i=0;i<forces.length;i++){
        if(Math.abs(forces[i].value)>Math.abs(highestForceValue)){
            highestForceValue=forces[i].value;
            highestForceIndex=i;
        }
    }

    // Replace the point and the high force value !!
    var highestMomentValue = 0.0;
    var highestMomentIndex = 0;
    for(var i=0;i<moments.length;i++){
        if(Math.abs(moments[i].value)>Math.abs(highestMomentValue)){
            highestMomentValue=moments[i].value;
            highestMomentIndex=i;
        }
    }

    // Find the point of the distribution !!
    var startDistX = Math.min(distributionStartX, distributionStartX+distributionLength);
    var endDistX = Math.max(distributionStartX, distributionStartX+distributionLength);

    // Find the left and right edge of the distribution !!
    var startDistY = getBeamYatX(startDistX);
    var endDistY = getBeamYatX(endDistX);
    if (startDistX == endDistX){
        startDistY = beamStartY;
        endDistY = beamStartY+beamLength*Math.sin(beamAngle);
        if(beamAngle==0.0&&beamLength==0.0){startDistY = mouseY;}
    }

    // Find the point of the distribution !!
    var highestDistValue = 0.0;
    var highestDistIndex = 0;
    for(var i=0;i<distributions.length;i++){
        if(Math.abs(distributions[i].value)>Math.abs(highestDistValue)){
            highestDistValue=distributions[i].value;
            highestDistIndex=i;
        }
    }

// Author of the Bending Moment !!
// Author of the Deflection !!
// Author of the Slope !!

// The Bending Moment Diagram Function !!
function drawBMD(){
    // Find the left and right edges of the beam
    var leftEdgeX = Math.min(beamStartX, beamStartX+beamLength);
    var rightEdgeX = Math.max(beamStartX, beamStartX+beamLength);

    // Find the left and right edge of the beam
    var leftEdgeY = getBeamYatX(leftEdgeX);
    var rightEdgeY = getBeamYatX(rightEdgeX);
    if (leftEdgeX == rightEdgeX){
        leftEdgeY = beamStartY;
        rightEdgeY = beamStartY+beamLength*Math.sin(beamAngle);
        if(beamAngle==0.0&&beamLength==0.0){leftEdgeY = mouseY;}
    }

    // Replace the point and the high force value !!
    var highestForceValue = 0.0;
    var highestForceIndex = 0;
    for(var i=0;i<forces.length;i++){
        if(Math.abs(forces[i].value)>Math.abs(highestForceValue)){
            highestForceValue=forces[i].value;
            highestForceIndex=i;
        }
    }

    // Replace the point and the high force value !!
    var highestMomentValue = 0.0;
    var highestMomentIndex = 0;
    for(var i=0;i<moments.length;i++){
        if(Math.abs(moments[i].value)>Math.abs(highestMomentValue)){
            highestMomentValue=moments[i].value;
            highestMomentIndex=i;
        }
    }

    // Find the point of the distribution !!
    var startDistX = Math.min(distributionStartX, distributionStartX+distributionLength);
    var endDistX = Math.max(distributionStartX, distributionStartX+distributionLength);

    // Find the left and right edge of the distribution !!
    var startDistY = getBeamYatX(startDistX);
    var endDistY = getBeamYatX(endDistX);
    if (startDistX == endDistX){
        startDistY = beamStartY;
        endDistY = beamStartY+beamLength*Math.sin(beamAngle);
        if(beamAngle==0.0&&beamLength==0.0){startDistY = mouseY;}
    }

    // Find the point of the distribution !!
    var highestDistValue = 0.0;
    var highestDistIndex = 0;
    for(var i=0;i<distributions.length;i++){
        if(Math.abs(distributions[i].value)>Math.abs(highestDistValue)){
            highestDistValue=distributions[i].value;
            highestDistIndex=i;
        }
    }

// Author of the Deflection !!
// Author of the Slope !!

// The Deflection Diagram Function !!
function drawDEF(){
    // Find the left and right edges of the beam
    var leftEdgeX = Math.min(beamStartX, beamStartX+beamLength);
    var rightEdgeX = Math.max(beamStartX, beamStartX+beamLength);

    // Find the left and right edge of the beam
    var leftEdgeY = getBeamYatX(leftEdgeX);
    var rightEdgeY = getBeamYatX(rightEdgeX);
    if (leftEdgeX == rightEdgeX){
        leftEdgeY = beamStartY;
        rightEdgeY = beamStartY+beamLength*Math.sin(beamAngle);
        if(beamAngle==0.0&&beamLength==0.0){leftEdgeY = mouseY;}
    }

    // Replace the point and the high force value !!
    var highestForceValue = 0.0;
    var highestForceIndex = 0;
    for(var i=0;i<forces.length;i++){
        if(Math.abs(forces[i].value)>Math.abs(highestForceValue)){
            highestForceValue=forces[i].value;
            highestForceIndex=i;
        }
    }

    // Replace the point and the high force value !!
    var highestMomentValue = 0.0;
    var highestMomentIndex = 0;
    for(var i=0;i<moments.length;i++){
        if(Math.abs(moments[i].value)>Math.abs(highestMomentValue)){
            highestMomentValue=moments[i].value;
            highestMomentIndex=i;
        }
    }

    // Find the point of the distribution !!
    var startDistX = Math.min(distributionStartX, distributionStartX+distributionLength);
    var endDistX = Math.max(distributionStartX, distributionStartX+distributionLength);

    // Find the left and right edge of the distribution !!
    var startDistY = getBeamYatX(startDistX);
    var endDistY = getBeamYatX(endDistX);
    if (startDistX == endDistX){
        startDistY = beamStartY;
        endDistY = beamStartY+beamLength*Math.sin(beamAngle);
        if(beamAngle==0.0&&beamLength==0.0){startDistY = mouseY;}
    }

    // Find the point of the distribution !!
    var highestDistValue = 0.0;
    var highestDistIndex = 0;
    for(var i=0;i<distributions.length;i++){
        if(Math.abs(distributions[i].value)>Math.abs(highestDistValue)){
            highestDistValue=distributions[i].value;
            highestDistIndex=i;
        }
    }

// Author of the Slope !!

// The Slope Diagram Function !!
function drawSLP(){
    // Find the left and right edges of the beam
    var leftEdgeX = Math.min(beamStartX, beamStartX+beamLength);
    var rightEdgeX = Math.max(beamStartX, beamStartX+beamLength);

    // Find the left and right edge of the beam
    var leftEdgeY = getBeamYatX(leftEdgeX);
    var rightEdgeY = getBeamYatX(rightEdgeX);
    if (leftEdgeX == rightEdgeX){
        leftEdgeY = beamStartY;
        rightEdgeY = beamStartY+beamLength*Math.sin(beamAngle);
        if(beamAngle==0.0&&beamLength==0.0){leftEdgeY = mouseY;}
    }

    // Replace the point and the high force value !!
    var highestForceValue = 0.0;
    var highestForceIndex = 0;
    for(var i=0;i<forces.length;i++){
        if(Math.abs(forces[i].value)>Math.abs(highestForceValue)){
            highestForceValue=forces[i].value;
            highestForceIndex=i;
        }
    }

    // Replace the point and the high force value !!
    var highestMomentValue = 0.0;
    var highestMomentIndex = 0;
    for(var i=0;i<moments.length;i++){
        if(Math.abs(moments[i].value)>Math.abs(highestMomentValue)){
            highestMomentValue=moments[i].value;
            highestMomentIndex=i;
        }
    }

    // Find the point of the distribution !!
    var startDistX = Math.min(distributionStartX, distributionStartX+distributionLength);
    var endDistX = Math.max(distributionStartX, distributionStartX+distributionLength);

    // Find the left and right edge of the distribution !!
    var startDistY = getBeamYatX(startDistX);
    var endDistY = getBeamYatX(endDistX);
    if (startDistX == endDistX){
        startDistY = beamStartY;
        endDistY = beamStartY+beamLength*Math.sin(beamAngle);
        if(beamAngle==0.0&&beamLength==0.0){startDistY = mouseY;}
    }

    // Find the point of the distribution !!
    var highestDistValue = 0.0;
    var highestDistIndex = 0;
    for(var i=0;i<distributions.length;i++){
        if(Math.abs(distributions[i].value)>Math.abs(highestDistValue)){
            highestDistValue=distributions[i].value;
            highestDistIndex=i;
        }
    }

// Author of the Shear Force Diagram !!
// Author of the Bending Moment !!
// Author of the Deflection !!
// Author of the Slope !!

// The Shear Force Diagram Function !!
function drawSFD(){
    // Find the left and right edges of the beam
    var leftEdgeX = Math.min(beamStartX, beamStartX+beamLength);
    var rightEdgeX = Math.max(beamStartX, beamStartX+beamLength);

    // Find the left and right edge of the beam
    var leftEdgeY = getBeamYatX(leftEdgeX);
    var rightEdgeY = getBeamYatX(rightEdgeX);
    if (leftEdgeX == rightEdgeX){
        leftEdgeY = beamStartY;
        rightEdgeY = beamStartY+beamLength*Math.sin(beamAngle);
        if(beamAngle==0.0&&beamLength==0.0){leftEdgeY = mouseY;}
    }

    // Replace the point and the high force value !!
    var highestForceValue = 0.0;
    var highestForceIndex = 0;
    for(var i=0;i<forces.length;i++){
        if(Math.abs(forces[i].value)>Math.abs(highestForceValue)){
            highestForceValue=forces[i].value;
            highestForceIndex=i;
        }
    }

    // Replace the point and the high force value !!
    var highestMomentValue = 0.0;
    var highestMomentIndex = 0;
    for(var i=0;i<moments.length;i++){
        if(Math.abs(moments[i].value)>Math.abs(highestMomentValue)){
            highestMomentValue=moments[i].value;
            highestMomentIndex=i;
        }
    }

    // Find the point of the distribution !!
    var startDistX = Math.min(distributionStartX, distributionStartX+distributionLength);
    var endDistX = Math.max(distributionStartX, distributionStartX+distributionLength);

    // Find the left and right edge of the distribution !!
    var startDistY = getBeamYatX(startDistX);
    var endDistY = getBeamYatX(endDistX);
    if (startDistX == endDistX){
        startDistY = beamStartY;
        endDistY = beamStartY+beamLength*Math.sin(beamAngle);
        if(beamAngle==0.0&&beamLength==0.0){startDistY = mouseY;}
    }

    // Find the point of the distribution !!
    var highestDistValue = 0.0;
    var highestDistIndex = 0;
    for(var i=0;i<distributions.length;i++){
        if(Math.abs(distributions[i].value)>Math.abs(highestDistValue)){
            highestDistValue=distributions[i].value;
            highestDistIndex=i;
        }
    }

// Author of the Bending Moment !!
// Author of the Deflection !!
// Author of the Slope !!

// The Bending Moment Diagram Function !!
function drawBMD(){
    // Find the left and right edges of the beam
    var leftEdgeX = Math.min(beamStartX, beamStartX+beamLength);
    var rightEdgeX = Math.max(beamStartX, beamStartX+beamLength);

    // Find the left and right edge of the beam
    var leftEdgeY = getBeamYatX(leftEdgeX);
    var rightEdgeY = getBeamYatX(rightEdgeX);
    if (leftEdgeX == rightEdgeX){
        leftEdgeY = beamStartY;
        rightEdgeY = beamStartY+beamLength*Math.sin(beamAngle);
        if(beamAngle==0.0&&beamLength==0.0){leftEdgeY = mouseY;}
    }

    // Replace the point and the high force value !!
    var highestForceValue = 0.0;
    var highestForceIndex = 0;
    for(var i=0;i<forces.length;i++){
        if(Math.abs(forces[i].value)>Math.abs(highestForceValue)){
            highestForceValue=forces[i].value;
            highestForceIndex=i;
        }
    }

    // Replace the point and the high force value !!
    var highestMomentValue = 0.0;
    var highestMomentIndex = 0;
    for(var i=0;i<moments.length;i++){
        if(Math.abs(moments[i].value)>Math.abs(highestMomentValue)){
            highestMomentValue=moments[i].value;
            highestMomentIndex=i;
        }
    }

    // Find the point of the distribution !!
    var startDistX = Math.min(distributionStartX, distributionStartX+distributionLength);
    var endDistX = Math.max(distributionStartX, distributionStartX+distributionLength);

    // Find the left and right edge of the distribution !!
    var startDistY = getBeamYatX(startDistX);
    var endDistY = getBeamYatX(endDistX);
    if (startDistX == endDistX){
        startDistY = beamStartY;
        endDistY = beamStartY+beamLength*Math.sin(beamAngle);
        if(beamAngle==0.0&&beamLength==0.0){startDistY = mouseY;}
    }

    // Find the point of the distribution !!
    var highestDistValue = 0.0;
    var highestDistIndex = 0;
    for(var i=0;i<distributions.length;i++){
        if(Math.abs(distributions[i].value)>Math.abs(highestDistValue)){
            highestDistValue=distributions[i].value;
            highestDistIndex=i;
        }
    }

// Author of the Deflection !!
// Author of the Slope !!

// The Deflection Diagram Function !!
function drawDEF(){
    // Find the left and right edges of the beam
    var leftEdgeX = Math.min(beamStartX, beamStartX+beamLength);
    var rightEdgeX = Math.max(beamStartX, beamStartX+beamLength);

    // Find the left and right edge of the beam
    var leftEdgeY = getBeamYatX(leftEdgeX);
    var rightEdgeY = getBeamYatX(rightEdgeX);
    if (leftEdgeX == rightEdgeX){
        leftEdgeY = beamStartY;
        rightEdgeY = beamStartY+beamLength*Math.sin(beamAngle);
        if(beamAngle==0.0&&beamLength==0.0){leftEdgeY = mouseY;}
    }

    // Replace the point and the high force value !!
    var highestForceValue = 0.0;
    var highestForceIndex = 0;
    for(var i=0;i<forces.length;i++){
        if(Math.abs(forces[i].value)>Math.abs(highestForceValue)){
            highestForceValue=forces[i].value;
            highestForceIndex=i;
        }
    }

    // Replace the point and the high force value !!
    var highestMomentValue = 0.0;
    var highestMomentIndex = 0;
    for(var i=0;i<moments.length;i++){
        if(Math.abs(moments[i].value)>Math.abs(highestMomentValue)){
            highestMomentValue=moments[i].value;
            highestMomentIndex=i;
        }
    }

    // Find the point of the distribution !!
    var startDistX = Math.min(distributionStartX, distributionStartX+distributionLength);
    var endDistX = Math.max(distributionStartX, distributionStartX+distributionLength);

    // Find the left and right edge of the distribution !!
    var startDistY = getBeamYatX(startDistX);
    var endDistY = getBeamYatX(endDistX);
    if (startDistX == endDistX){
        startDistY = beamStartY;
        endDistY = beamStartY+beamLength*Math.sin(beamAngle);
        if(beamAngle==0.0&&beamLength==0.0){startDistY = mouseY;}
    }

    // Find the point of the distribution !!
    var highestDistValue = 0.0;
    var highestDistIndex = 0;
    for(var i=0;i<distributions.length;i++){
        if(Math.abs(distributions[i].value)>Math.abs(highestDistValue)){
            highestDistValue=distributions[i].value;
            highestDistIndex=i;
        }
    }

// Author of the Slope !!

// The Slope Diagram Function !!
function drawSLP(){
    // Find the left and right edges of the beam
    var leftEdgeX = Math.min(beamStartX, beamStartX+beamLength);
    var rightEdgeX = Math.max(beamStartX, beamStartX+beamLength);

    // Find the left and right edge of the beam
    var leftEdgeY = getBeamYatX(leftEdgeX);
    var rightEdgeY = getBeamYatX(rightEdgeX);
    if (leftEdgeX == rightEdgeX){
        leftEdgeY = beamStartY;
        rightEdgeY = beamStartY+beamLength*Math.sin(beamAngle);
        if(beamAngle==0.0&&beamLength==0.0){leftEdgeY = mouseY;}
    }

    // Replace the point and the high force value !!
    var highestForceValue = 0.0;
    var highestForceIndex = 0;
    for(var i=0;i<forces.length;i++){
        if(Math.abs(forces[i].value)>Math.abs(highestForceValue)){
            highestForceValue=forces[i].value;
            highestForceIndex=i;
        }
    }

    // Replace the point and the high force value !!
    var highestMomentValue = 0.0;
    var highestMomentIndex = 0;
    for(var i=0;i<moments.length;i++){
        if(Math.abs(moments[i].value)>Math.abs(highestMomentValue)){
            highestMomentValue=moments[i].value;
            highestMomentIndex=i;
        }
    }

    // Find the point of the distribution !!
    var startDistX = Math.min(distributionStartX, distributionStartX+distributionLength);
    var endDistX = Math.max(distributionStartX, distributionStartX+distributionLength);

    // Find the left and right edge of the distribution !!
    var startDistY = getBeamYatX(startDistX);
    var endDistY = getBeamYatX(endDistX);
    if (startDistX == endDistX){
        startDistY = beamStartY;
        endDistY = beamStartY+beamLength*Math.sin(beamAngle);
        if(beamAngle==0.0&&beamLength==0.0){startDistY = mouseY;}
    }

    // Find the point of the distribution !!
    var highestDistValue = 0.0;
    var highestDistIndex = 0;
    for(var i=0;i<distributions.length;i++){
        if(Math.abs(distributions[i].value)>Math.abs(highestDistValue)){
            highestDistValue=distributions[i].value;
            highestDistIndex=i;
        }
    }

// Author of the Shear Force Diagram !!
// Author of the Bending Moment !!
// Author of the Deflection !!
// Author of the Slope !!

// The Shear Force Diagram Function !!
function drawSFD(){
    // Find the left and right edges of the beam
    var leftEdgeX = Math.min(beamStartX, beamStartX+beamLength);
    var rightEdgeX = Math.max(beamStartX, beamStartX+beamLength);

    // Find the left and right edge of the beam
    var leftEdgeY = getBeamYatX(leftEdgeX);
    var rightEdgeY = getBeamYatX(rightEdgeX);
    if (leftEdgeX == rightEdgeX){
        leftEdgeY = beamStartY;
        rightEdgeY = beamStartY+beamLength*Math.sin(beamAngle);
        if(beamAngle==0.0&&beamLength==0.0){leftEdgeY = mouseY;}
    }

    // Replace the point and the high force value !!
    var highestForceValue = 0.0;
    var highestForceIndex = 0;
    for(var i=0;i<forces.length;i++){
        if(Math.abs(forces[i].value)>Math.abs(highestForceValue)){
            highestForceValue=forces[i].value;
            highestForceIndex=i;
        }
    }

    // Replace the point and the high force value !!
    var highestMomentValue = 0.0;
    var highestMomentIndex = 0;
    for(var i=0;i<moments.length;i++){
        if(Math.abs(moments[i].value)>Math.abs(highestMomentValue)){
            highestMomentValue=moments[i].value;
            highestMomentIndex=i;
        }
    }

    // Find the point of the distribution !!
    var startDistX = Math.min(distributionStartX, distributionStartX+distributionLength);
    var endDistX = Math.max(distributionStartX, distributionStartX+distributionLength);

    // Find the left and right edge of the distribution !!
    var startDistY = getBeamYatX(startDistX);
    var endDistY = getBeamYatX(endDistX);
    if (startDistX == endDistX){
        startDistY = beamStartY;
        endDistY = beamStartY+beamLength*Math.sin(beamAngle);
        if(beamAngle==0.0&&beamLength==0.0){startDistY = mouseY;}
    }

    // Find the point of the distribution !!
    var highestDistValue = 0.0;
    var highestDistIndex = 0;
    for(var i=0;i<distributions.length;i++){
        if(Math.abs(distributions[i].value)>Math.abs(highestDistValue)){
            highestDistValue=distributions[i].value;
            highestDistIndex=i;
        }
    }

// Author of the Bending Moment !!
// Author of the Deflection !!
// Author of the Slope !!

// The Bending Moment Diagram Function !!
function drawBMD(){
    // Find the left and right edges of the beam
    var leftEdgeX = Math.min(beamStartX, beamStartX+beamLength);
    var rightEdgeX = Math.max(beamStartX, beamStartX+beamLength);

    // Find the left and right edge of the beam
    var leftEdgeY = getBeamYatX(leftEdgeX);
    var rightEdgeY = getBeamYatX(rightEdgeX);
    if (leftEdgeX == rightEdgeX){
        leftEdgeY = beamStartY;
        rightEdgeY = beamStartY+beamLength*Math.sin(beamAngle);
        if(beamAngle==0.0&&beamLength==0.0){leftEdgeY = mouseY;}
    }

    // Replace the point and the high force value !!
    var highestForceValue = 0.0;
    var highestForceIndex = 0;
    for(var i=0;i<forces.length;i++){
        if(Math.abs(forces[i].value)>Math.abs(highestForceValue)){
            highestForceValue=forces[i].value;
            highestForceIndex=i;
        }
    }

    // Replace the point and the high force value !!
    var highestMomentValue = 0.0;
    var highestMomentIndex = 0;
    for(var i=0;i<moments.length;i++){
        if(Math.abs(moments[i].value)>Math.abs(highestMomentValue)){
            highestMomentValue=moments[i].value;
            highestMomentIndex=i;
        }
    }

    // Find the point of the distribution !!
    var startDistX = Math.min(distributionStartX, distributionStartX+distributionLength);
    var endDistX = Math.max(distributionStartX, distributionStartX+distributionLength);

    // Find the left and right edge of the distribution !!
    var startDistY = getBeamYatX(startDistX);
    var endDistY = getBeamYatX(endDistX);
    if (startDistX == endDistX){
        startDistY = beamStartY;
        endDistY = beamStartY+beamLength*Math.sin(beamAngle);
        if(beamAngle==0.0&&beamLength==0.0){startDistY = mouseY;}
    }

    // Find the point of the distribution !!
    var highestDistValue = 0.0;
    var highestDistIndex = 0;
    for(var i=0;i<distributions.length;i++){
        if(Math.abs(distributions[i].value)>Math.abs(highestDistValue)){
            highestDistValue=distributions[i].value;
            highestDistIndex=i;
        }
    }

// Author of the Deflection !!
// Author of the Slope !!

// The Deflection Diagram Function !!
function drawDEF(){
    // Find the left and right edges of the beam
    var leftEdgeX = Math.min(beamStartX, beamStartX+beamLength);
    var rightEdgeX = Math.max(beamStartX, beamStartX+beamLength);

    // Find the left and right edge of the beam
    var leftEdgeY = getBeamYatX(leftEdgeX);
    var rightEdgeY = getBeamYatX(rightEdgeX);
    if (leftEdgeX == rightEdgeX){
        leftEdgeY = beamStartY;
        rightEdgeY = beamStartY+beamLength*Math.sin(beamAngle);
        if(beamAngle==0.0&&beamLength==0.0){leftEdgeY = mouseY;}
    }

    // Replace the point and the high force value !!
    var highestForceValue = 0.0;
    var highestForceIndex = 0;
    for(var i=0;i<forces.length;i++){
        if(Math.abs(forces[i].value)>Math.abs(highestForceValue)){
            highestForceValue=forces[i].value;
            highestForceIndex=i;
        }
    }

    // Replace the point and the high force value !!
    var highestMomentValue = 0.0;
    var highestMomentIndex = 0;
    for(var i=0;i<moments.length;i++){
        if(Math.abs(moments[i].value)>Math.abs(highestMomentValue)){
            highestMomentValue=moments[i].value;
            highestMomentIndex=i;
        }
    }

    // Find the point of the distribution !!
    var startDistX = Math.min(distributionStartX, distributionStartX+distributionLength);
    var endDistX = Math.max(distributionStartX, distributionStartX+distributionLength);

    // Find the left and right edge of the distribution !!
    var startDistY = getBeamYatX(startDistX);
    var endDistY = getBeamYatX(endDistX);
    if (startDistX == endDistX){
        startDistY = beamStartY;
        endDistY = beamStartY+beamLength*Math.sin(beamAngle);
        if(beamAngle==0.0&&beamLength==0.0){startDistY = mouseY;}
    }

    // Find the point of the distribution !!
    var highestDistValue = 0.0;
    var highestDistIndex = 0;
    for(var i=0;i<distributions.length;i++){
        if(Math.abs(distributions[i].value)>Math.abs(highestDistValue)){
            highestDistValue=distributions[i].value;
            highestDistIndex=i;
        }
    }

// Author of the Slope !!

// The Slope Diagram Function !!
function drawSLP(){
    // Find the left and right edges of the beam
    var leftEdgeX = Math.min(beamStartX, beamStartX+beamLength);
    var rightEdgeX = Math.max(beamStartX, beamStartX+beamLength);

    // Find the left and right edge of the beam
    var leftEdgeY = getBeamYatX(leftEdgeX);
    var rightEdgeY = getBeamYatX(rightEdgeX);
    if (leftEdgeX == rightEdgeX){
        leftEdgeY = beamStartY;
        rightEdgeY = beamStartY+beamLength*Math.sin(beamAngle);
        if(beamAngle==0.0&&beamLength==0.0){leftEdgeY = mouseY;}
    }

    // Replace the point and the high force value !!
    var highestForceValue = 0.0;
    var highestForceIndex = 0;
    for(var i=0;i<forces.length;i++){
        if(Math.abs(forces[i].value)>Math.abs(highestForceValue)){
            highestForceValue=forces[i].value;
            highestForceIndex=i;
        }
    }

    // Replace the point and the high force value !!
    var highestMomentValue = 0.0;
    var highestMomentIndex = 0;
    for(var i=0;i<moments.length;i++){
        if(Math.abs(moments[i].value)>Math.abs(highestMomentValue)){
            highestMomentValue=moments[i].value;
            highestMomentIndex=i;
        }
    }

    // Find the point of the distribution !!
    var startDistX = Math.min(distributionStartX, distributionStartX+distributionLength);
    var endDistX = Math.max(distributionStartX, distributionStartX+distributionLength);

    // Find the left and right edge of the distribution !!
    var startDistY = getBeamYatX(startDistX);
    var endDistY = getBeamYatX(endDistX);
    if (startDistX == endDistX){
        startDistY = beamStartY;
        endDistY = beamStartY+beamLength*Math.sin(beamAngle);
        if(beamAngle==0.0&&beamLength==0.0){startDistY = mouseY;}
    }

    // Find the point of the distribution !!
    var highestDistValue = 0.0;
    var highestDistIndex = 0;
    for(var i=0;i<distributions.length;i++){
        if(Math.abs(distributions[i].value)>Math.abs(highestDistValue)){
            highestDistValue=distributions[i].value;
            highestDistIndex=i;
        }
    }

// The Shear Force Diagram !!
// The Bending Moment Diagram !!
// The Deflection Diagram !!
// The Slope Diagram !!

// The Shear Force Diagram !!
var shearForceDiagram = [];
var shearForceDiagramLength = 0;
var shearForceDiagramX = [];
var shearForceDiagramY = [];
var shearForceDiagramLeftEdgeX = 0.0;
var shearForceDiagramLeftEdgeY = 0.0;
var shearForceDiagramRightEdgeX = 0.0;
var shearForceDiagramRightEdgeY = 0.0;

// The Bending Moment Diagram !!
var bendingMomentDiagram = [];
var bendingMomentDiagramLength = 0;
var bendingMomentDiagramX = [];
var bendingMomentDiagramY = [];
var bendingMomentDiagramLeftEdgeX = 0.0;
var bendingMomentDiagramLeftEdgeY = 0.0;
var bendingMomentDiagramRightEdgeX = 0.0;
var bendingMomentDiagramRightEdgeY = 0.0;

// The Deflection Diagram !!
var deflectionDiagram = [];
var deflectionDiagramLength = 0;
var deflectionDiagramX = [];
var deflectionDiagramY = [];
var deflectionDiagramLeftEdgeX = 0.0;
var deflectionDiagramLeftEdgeY = 0.0;
var deflectionDiagramRightEdgeX = 0.0;
var deflectionDiagramRightEdgeY = 0.0;

// The Slope Diagram !!
var slopeDiagram = [];
var slopeDiagramLength = 0;
var slopeDiagramX = [];
var slopeDiagramY = [];
var slopeDiagramLeftEdgeX = 0.0;
var slopeDiagramLeftEdgeY = 0.0;
var slopeDiagramRightEdgeX = 0.0;
var slopeDiagramRightEdgeY = 0.0;

// Author of the Shear Force Diagram !!
// Author of the Bending Moment !!
// Author of the Deflection !!
// Author of the Slope !!

// The Shear Force Diagram !!
var shearForceDiagram = [];
var shearForceDiagramLength = 0;
var shearForceDiagramX = [];
var shearForceDiagramY = [];
var shearForceDiagramLeftEdgeX = 0.0;
var shearForceDiagramLeftEdgeY = 0.0;
var shearForceDiagramRightEdgeX = 0.0;
var shearForceDiagramRightEdgeY = 0.0;

// The Bending Moment Diagram !!
var bendingMomentDiagram = [];
var bendingMomentDiagramLength = 0;
var bendingMomentDiagramX = [];
var bendingMomentDiagramY = [];
var bendingMomentDiagramLeftEdgeX = 0.0;
var bendingMomentDiagramLeftEdgeY = 0.0;
var bendingMomentDiagramRightEdgeX = 0.0;
var bendingMomentDiagramRightEdgeY = 0.0;

// The Deflection Diagram !!
var deflectionDiagram = [];
var deflectionDiagramLength = 0;
var deflectionDiagramX = [];
var deflectionDiagramY = [];
var deflectionDiagramLeftEdgeX = 0.0;
var deflectionDiagramLeftEdgeY = 0.0;
var deflectionDiagramRightEdgeX = 0.0;
var deflectionDiagramRightEdgeY = 0.0;

// The Slope Diagram !!
var slopeDiagram = [];
var slopeDiagramLength = 0;
var slopeDiagramX = [];
var slopeDiagramY = [];
var slopeDiagramLeftEdgeX = 0.0;
var slopeDiagramLeftEdgeY = 0.0;
var slopeDiagramRightEdgeX = 0.0;
var slopeDiagramRightEdgeY = 0.0;
```

```
  ( append a message to show the output box )
  (setf (gtk:text-view-buffer output-view) (make-instance :text-buffer :text (f ...))
        (gtk:text-view-cursor-visible output-view) nil
        (gtk:text-view-editable output-view) nil
        (gtk:text-view-wrap-mode output-view) :word)
  )
)

;; Keep the root alive.
(defvar *root-box* nil)
(setf *root-box* (gtk:gtk-window-new :type :toplevel :title "Lisp Calculator"
                  :default-height 600 :default-width 800 :window-position :center-on-parent))
(g:g-signal-connect *root-box* "destroy"
                    (lambda (w) (declare (ignore w)) (setf *root-box* nil) (gtk:leave-gtk-main) t))
(gtk:gtk-widget-show-all *root-box*)
(gtk:gtk-main)
```

```
;; Load the GTK bindings.
(ql:quickload :cl-cairo2 :silent t)
(ql:quickload :cl-cairo2-x11 :silent t)
(ql:quickload :cl-pango :silent t)
(ql:quickload :cl-pangocairo :silent t)
(ql:quickload :cl-gtk2-gtk :silent t
```

```
// Apply drawing to the image.
// TODO: Implement drawing.

// Get the bounds.
bounds = image.Bounds()

// Create the new image.
newImage = image.NewRGBA(image.Rectangle{ Min: pt, Max: pt.Add(bounds.Dx(), bounds.Dy()) })

// Set the pixels.
for x := bounds.Min.X; x < bounds.Max.X; x++ {
    for y := bounds.Min.Y; y < bounds.Max.Y; y++ {
        newImage.Set(x, y, image.At(x, y))
    }
}

// Create the PNG encoder.
encoder := &png.Encoder{ CompressionLevel: png.DefaultCompression, BufferPool: nil }

// Save out to the new file.
if err = encoder.Encode(outfile, newImage); err != nil {
    fmt.Printf("Error encoding the image: %v\n", err)
    os.Exit(1)
}

// Write the success message.
fmt.Printf("Successfully wrote %d x %d image to %s\n", bounds.Dx(), bounds.Dy(), outfile.Name())

// Exit with success.
os.Exit(0)
}

// Parse the command line.
func parseCommandLine() (*os.File, *os.File, int, int, int
```