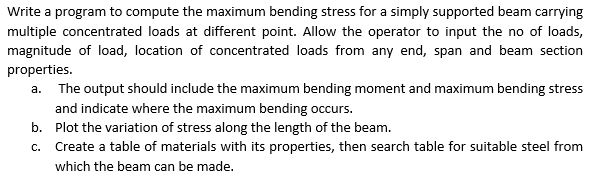
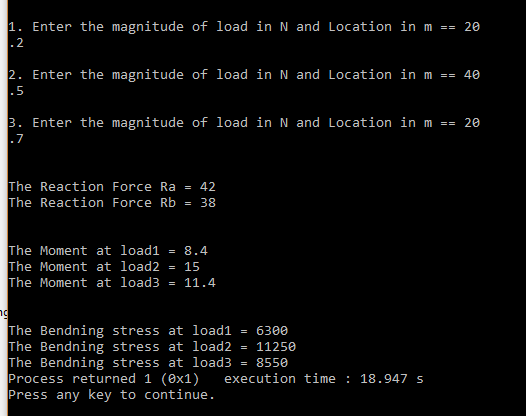
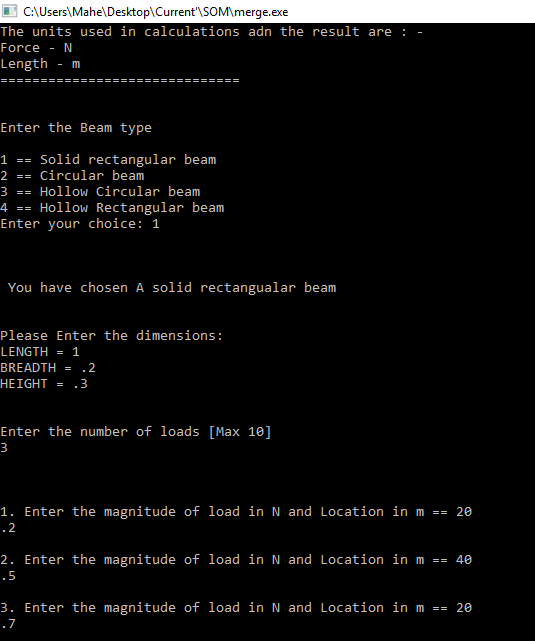
SOM

PROJECT

Problem. Statement

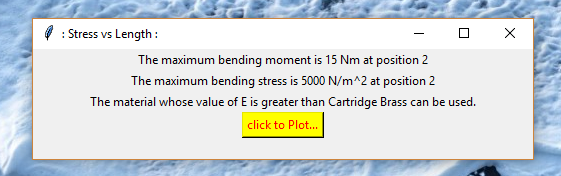


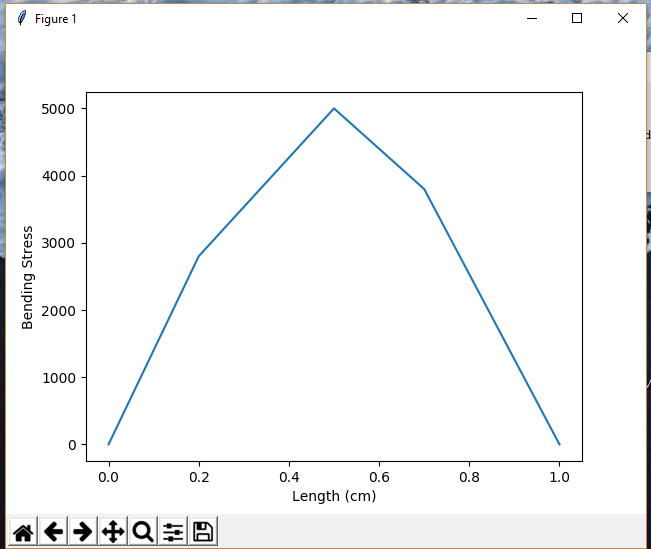
Screenshots of the Program

Enter the choice

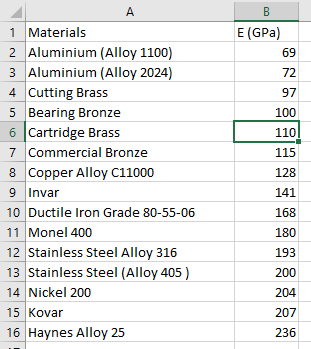
Enter the dimensions and no of loads

The Output





The Graph

The selected materials

The main logic behind the code.

void Rect::moments\_finder(double get[])

{

double a[no\_of\_load\_var];

for(int i=0 ; i<no\_of\_load\_var;i++)

{

a[i]=0;

get[i]=0;

}

get[0]=ra\*load\_location[0][1];

for(int i =1 ; i<no\_of\_load\_var;i++)

{

for(int j =0 ; j<i;j++)

{

a[i]=a[i]+(load\_location[j][0])\*(load\_location[i][1]-load\_location[j][1]);

get[i]=(ra\*load\_location[i][1])-a[i];

}

}

void Rect::bending\_stress(double get\_stresses [],double moments[])

{

ofstream file;

file.open ("out.txt");

file<<"0"<<","<<"0"<<"\n";

for(int i = 0; i<no\_of\_load\_var;i++)

{

get\_stresses[i]=(moments[i]\*0.5\*h)/(Inertia());

std::cout<<"\nThe Bendning stress at load"<<i+1<<" = "<<get\_stresses[i];

file<<get\_stresses[i]<<","<<load\_location[i][1]<<"\n";

}

file<<"0"<<","<<(L)<<"\n";

file.close();

for(int i =0 ; i<no\_of\_load\_var;i++)

{

if(max<moments[i])

{

max = moments[i];

pos=i;

}

}

ofstream abc;

abc.open("som.txt");

abc<<max<<" "<<Inertia()<<" "<<h<<" "<<get\_stresses[pos]<<" "<<pos+1;

abc.close();

}