**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**BELAGAVI-590018, KARNATAKA**

**A picture containing diagram

Description automatically generated**

A Mini Project Report on

**“STACK IMPLEMENTATION”**

Mini Project Report submitted in partial fulfillment of the requirement for the CG Laboratory with Mini Project [18CSL68]

**Bachelor of Engineering**

**In**

**Computer Science and Engineering**

**Submitted By**

**Aditya Krishnan**

**1JT19CS004**

**Abhishek Kumar**

**1JT19CS002**

**A picture containing text, clipart

Description automatically generated**

**Department of Computer Science and Engineering**

**Jyothy Institute of Technology,**

**Tataguni, Bengaluru – 560082**

**Department of Computer Science and Engineering**

**Jyothy Institute of Technology,**

A picture containing text, clipart

Description automatically generated**Tataguni, Bengaluru – 560082**

**CERTIFICATE**

Certified that the mini project work entitled **“STACK IMPLEMENTATION”** carried out by **Aditya Krishnan [1JT19CS004] and Abhishek Kumar [1JT19CS002]** bonafide students of Jyothy Institute of Technology, in partial fulfillment for the award of **Bachelor of Engineering** in **Computer Science and Engineering** department of the **Visvesvaraya Technological University, Belagavi** during the year **2021-2022**. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree

###### **Mrs. Roopa Deshpande Dr.Prabhanjan S**

###### **Guide, Asst. Professor Professor & HOD**

Dept. of CSE Dept. of CSE

External Viva Examiner Signature with Date:

1.

2.

**ACKNOWLEDGEMENT**

Firstly, we are very grateful to this esteemed institution **“Jyothy Institute of Technology”** for providing us an opportunity to complete our project.

We express our sincere thanks to our **Principal Dr. Gopalakrishna K** for providing us with adequate facilities to undertake this project.

We would like to thank **Dr. Prabhanjan S**, **Professor and Head of Computer Science and Engineering Department** for providing his valuable support.

We would like to thank our guides **Mrs. Roopa Deshpande, Assistant Professor** for their keen interest and guidance in preparing this work.

Finally, we would thank all our friends who have helped us directly or indirectly in this project.

**Aditya Krishnan [1JT19CS004]**

**Abhishek Kumar [1JT19CS002]**

**ABSTRACT**

The aim of this project is to Perform a “Stack Operation”. The objective of this project is to provide a visual implementation of the functioning of a stack. Users can push and pop elements into the stack, and the result is drawn on the screen using computer graphics. The graphics API that is used are OpenGL libraries and GLUT.

A stack is a conceptual structure consisting of a set of homogeneous elements and is based on the principle of last in first out (LIFO). It is a commonly used abstract data type with two major operations, namely, push and pop. Push and pop are carried out on the topmost element, which is the item most recently added to the stack. The push operation adds an element to the stack while the pop operation removes an element from the top position. The stack concept is used in programming and memory organization in computers.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **Sl No** | **Description** | **Page No** |
| 1 | INTRODUCTION | 6 |
| 2 | REQUIREMENT ANALYSIS | 10 |
| 3 | IMPLEMENTATION | 12 |
| 4 | SNAPSHOTS | 14 |
| 5 | CONCLUSION | 18 |
| 6 | REFERENCES | 19 |
| 7 | APPENDIX | 20 |

**CHAPTER 1**

**INTRODUCTION**

**1.1 Introduction to Computer Graphics**

Computer Graphics are defined as any sketch or a drawing or a special network that pictorially represents some meaningful information. Computer Graphics is used where a set of images needs to be manipulated or the creation of the image in the form of pixels and is drawn on the computer. Computer Graphics can be used in digital photography, film, entertainment, electronic gadgets, and all other core technologies which are required. It is a vast subject and area in the field of computer science. Computer Graphics can be used in UI design, rendering, geometric objects, animation, and many more. In most areas, computer graphics is an abbreviation of CG. There are several tools used for the implementation of Computer Graphics. The basic is the <graphics.h> header file in Turbo-C, Unity for advanced, and even OpenGL can be used for its Implementation. It was invented in 1960 by great researchers Verne Hudson and William Fetter from Boeing.

**Computer Graphics refers to several things:**

* The manipulation and the representation of the image or the data in a graphical manner.
* Various technology is required for the creation and manipulation.
* Digital synthesis and its manipulation.

**Advantages of Computer Graphics:**

* Increases Usability
* Product Development and Research
* Employment Opportunities
* Designing
* Teaching Learning activities becomes easy

**1.2 Applications of Computer Graphics**

The technology that dealt with designs and images on a computer screen are the computer graphics. Computer graphics are visual representations of data shown on a computer monitor. Computer graphics are used for the creation, development of videos and computer programs, science modelling, catalog design and other commercial art. Today almost every machine is able to do some graphics, and people have even expected to control their computer through icons and images instead of simply typing.

**ComputerArt:**  
Using computer graphics we can create fine and commercial art which include animation packages, paint packages. These packages provide facilities for designing object shapes and specifying object motion.Cartoon drawing, paintings, logo design can also be done.

**Computer-Aided-Drawing:**  
Designing of buildings, automobile, aircraft is done with the help of computer aided drawing, this helps in providing minute details to the drawing and producing more accurate and sharp drawings with better specifications.

**Presentation-Graphics:**  
For the preparation of reports or summarising the financial, statistical, mathematical, scientific, economic data for research reports, managerial reports, moreover creation of bar graphs, pie charts, time chart, can be done using the tools present in computer graphics.

**Entertainment:**  
Computer graphics finds a major part of its utility in the movie industry and game industry. Used for creating motion pictures , music video, television shows, cartoon animation films. In the game industry where focus and interactivity are the key players, computer graphics helps in providing such features in the efficient way.

**Training:**  
Specialised system for training like simulators can be used for training the candidates in a way that can be grasped in a short span of understanding. Creation of training modules using computer graphics is simple and very useful.

**Visualisation:**  
Today the need of visualise things have increased drastically, the need of visualisation can be seen in many advance technologies, data visualisation helps in finding insights of the data , to check and study the behaviour of processes around us we need appropriate visualisation which can be achieved through proper usage of computer graphics

**Image-Processing:**  
Various kinds of photographs or images require editing in order to be used in different places. Processing of existing images into refined ones for better interpretation is one of the many applications of computer graphics.

**Machine-Drawing:**  
Computer graphics is very frequently used for designing, modifying and creation of various parts of machine and the whole machine itself, the main reason behind using computer graphics for this purpose is the precision and clarity we get from such drawing is ultimate and extremely desired for the safe manufacturing of machine using these drawing

**1.3 History of OpenGL**

The Open Graphics Library (OpenGL) is a standard specification defining a cross-language, cross-platform Application Program(ming) Interface (API). OpenGL provides the programmer an interface to the graphics hardware. OpenGL is a powerful, low-level rendering and modeling software library. With OpenGL, it is possible to produce 2D and 3D graphics applications. OpenGL is widely used in video games, CAD, scientific applications, etc, on many different platforms.

In the 1980's developing software was a challenge. Especially when you wanted to program for a wide range of graphics hardware. (Remember, at that time there were no API's that could be used).

The software developers had to write custom drivers for each piece of hardware. They also had to deal with a lot of different interfaces, which made the task even harder. (The development cycle was slow because of it). Another problem was that each development team (from different companies) had to write drivers for the same piece of hardware. The result was that there was a lot of the same source code been written by different teams. (Waste of time and money).

In the early 90's Silicon Graphics inc. was a leader in 3D graphics for workstations. They used an API called IRIS GL for their workstations. IRIS GL was proprietary to SGI's hardware and not an "open" standard. The API was considered too easy to use and it also supported immediate mode rendering.

At that time competing vendors, including Sun Microsystems, IBM and Hewlett-Packard were also bringing 3D hardware on the market. They used another API called PHIGS. (In functionality and capability IRIS GL was superior to PHIGS).

Because other vendors brought new 3D hardware on the market, SGI's market share became smaller.

To turn the tide and influence the market SGI decided to turn the IRIS GL into an open standard. (They could not make IRIS GL an open standard because of licensing and patent issues). They made a new API based on IRIS GL called OpenGL.

In 1992, SGI led the creation of the OpenGL architectural review board (ARB). (The founding companies of the ARB were: SGI, Microsoft, IBM, DEC and Intel. Today nine companies have voting seats on the ARB, and several more attend the quarterly meetings to provide input to the evolution of OpenGL). The role of the OpenGL ARB is to establish and maintain the OpenGL specifications.

Not all advanced hardware-specific features can be accessed by the OpenGL versions (Default OpenGL). Fortunately, the video-card manufactures can and do provide OpenGL extensions. With these extensions you are able to access advanced hardware-specific features.

If these features are used by many vendors, the extensions can become an official addition to the OpenGL standard. Future releases of OpenGL will be more and more influenced by game developers and the gaming industry.

**CHAPTER 2**

**REQUIREMENT ANALYSIS**

**2.1 Domain Understanding**

The main objective is to develop a suitable OpenGL graphics package to implement basic computer graphic skills. The aim of this project is to provide a visual representation of the functioning of a Stack using computer graphics. Classification of Requirements User Requirements Program to demonstrate the inner working of a stack using computer graphics.

**2.2 Requirement Collection**

The objects used in this project are rectangles that represent the elements in a stack, as well as the stack itself. Additionally, the text is used to describe the stack, whether there is an overflow or underflow.

**2.3** **User Defined functions**

void display()

void myinit()

void keys()

void main(int argc, char\*\* argv).

**2.4 System Requirements**

We are using the GL/glut.h header file to implement OpenGL functionality in the program.The OpenGL Utility Toolkit (GLUT) is a library of utilities for OpenGL programs, which primarily perform system-level I/O with the host operating system. Functions performed include window definition, window control, and monitoring of keyboard and mouse input. Additionally we use stdlib.h, string.h and stdio.h to add basic input out functions to our program.

**2.5 Software Requirements**

Operating System - Ubuntu

Programming Language - C

Graphics API - OpenGL

**2.6 Hardware Requirements**

Processor: Pentium Processor

Memory: 32MB Ram

Hard Disk: 40GB Hard Disk

Interface Devices: Mouse, Keyboard and Monitor

**CHAPTER 3**

**IMPLEMENTATION**

**3.1 Description of Implementation Module**

In this project I have created a game using “OpenGL” Function API by the help of the built in function present in the headerfile. To provide functionality to our project in header file.We have used subfunctions. The subfunctions provide us efficient way to design the project.In this chapter we are describing the functionality of our project using functions.

**3.2 List of Implementation of the Functions**

* void Init()
* void display()
* void keys(unsigned char key,int x,int y)
* void main(int argc, char\*\* argv)

**3.2 Description of the Functions**

main(): The execution of the program starts from the main.

glutInit(): It is used to initialize the GLUT library. The arguments are passed in and used by the applications.

glutInitWindowSize(): It specifies the initial size of the window which consists of the height and width of the window in the pixels.

glutCreateWindow(): It will create a window on the display. The string can be used to the label the window.

glutDisplayFun(): It registers the display function that is executed when the window needs to redraw.

glutMainLoop(): It causes the program to enter an event processing loop.

myInit(): It is defined to initialize the window parameters.

glutIdleFunc(): It sets the global idle callback to be function so a GLUT program can perform background processing tasks or continuous animation when window system events are not being received.

glutKeyboardFunc():It sets the keyboard callback for the current Window.When a user types into the window, each key press generating an ASCII character will generate a keyboard callback.

**3.3 Stack Implementation**

A stack is a conceptual structure consisting of a set of homogeneous elements and is based on the principle of last in first out (LIFO). It is a commonly used abstract data type with two major operations, namely push and pop. Push and pop are carried out on the topmost element, which is the item most recently added to the stack. The push operation adds an element to the stack while the pop operation removes an element from the top position. The stack concept is used in programming and memory organization in computers.

**3.4 Stack Algorithm**

int counter=0; counter initialization

if (key==’p’)

Counter++

increase counter by one and push the element into the stack

if(key==’o’)

decrease the counter by one and pop the element from the stack.

if (counter =<0)

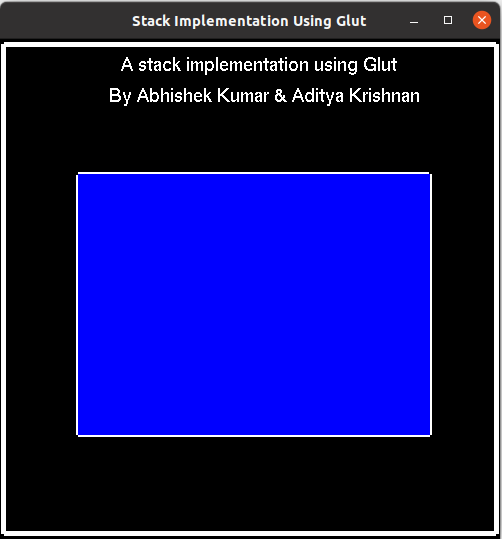
Stack is empty

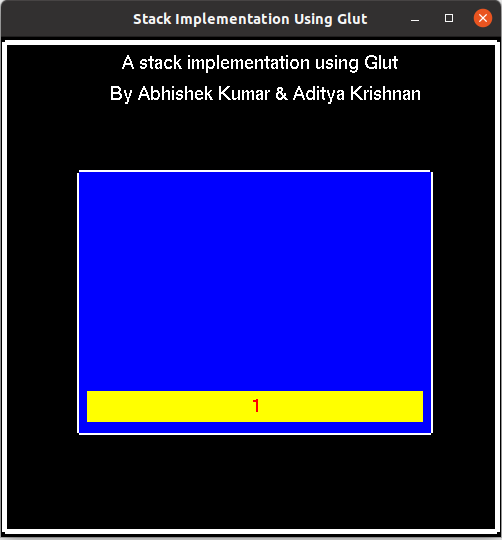
if (counter >=5)

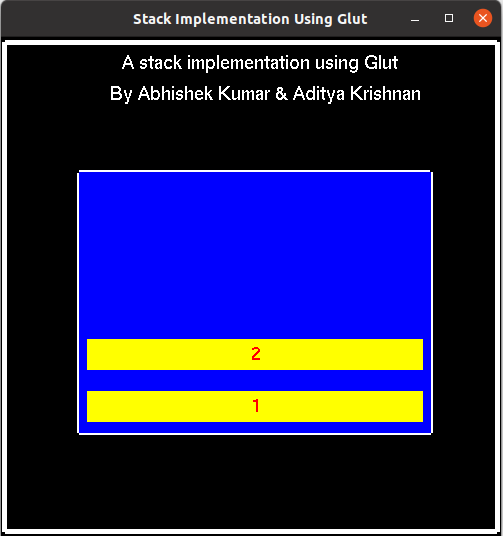
Stack is full

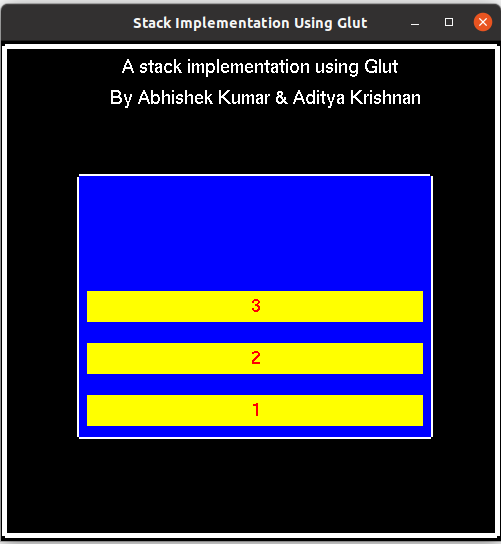
**CHAPTER 4**

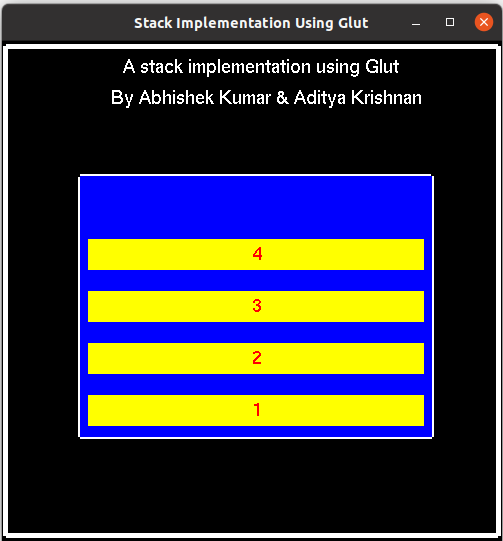
**SNAPSHOTS**

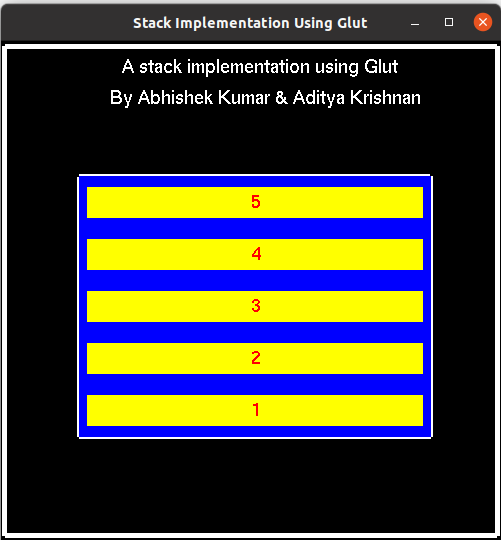
****

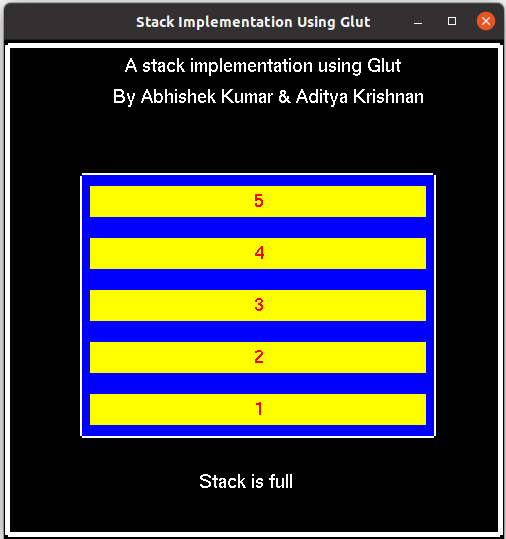
****

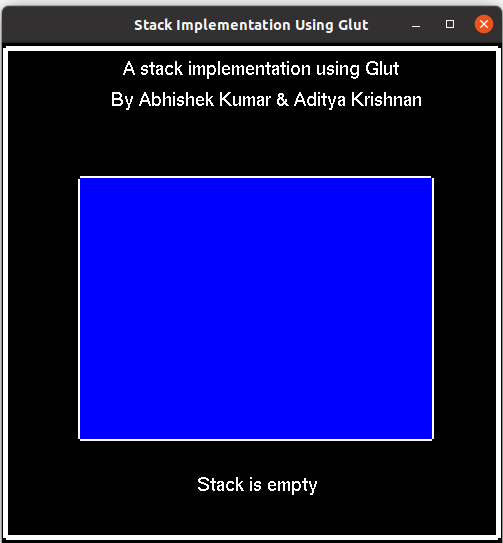
****

****

****

****





**CHAPTER 5**

**CONCLUSION**

This mini-project makes it easy for students to get an understanding of how a stack works by following the push and pop operations that are being performed on the stack. It is also easy to understand why a stack underflows or overflows and the measures taken to prevent such errors from occurring.

**REFERENCES**

[1] The OpenGL Utility Toolkit (GLUT) documentation:

<https://www.opengl.org/resources/libraries/glut/spec3/spec3.html/>  
[2] Donald D. Hearn, M. Pauline Baker -Computer Graphics with OpenGL (3rd Edition)-Prentice Hall (2003)

[3] Edward Angel- Interactive Computer Graphics Atop- Down Approach Using OpenGL (5th Edition) (2008)

Interactive Computer Graphics: A Top-Down Approach using OpenGL by Edward Angel

**APPENDIX**

.

#include<GL/glut.h>

#include<stdlib.h>

#include<stdio.h>

#include<string.h>

int counter = 0;

void myInit(void)

{

glClearColor(0.0,0.0,0.0,4.0);

glColor3f(0.0f,0.0f,0.0f);

glPointSize(4.0);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluOrtho2D(0.0,640.0,0.0,480.0);

}

void display(void)

{

glClear(GL\_COLOR\_BUFFER\_BIT);

glColor3f( 1, 1, 1 );

// glRasterPos2f(275,450);

glRasterPos2f(155,450);

int len, i;

char string[]="A stack implementation using Glut";

len = (int)strlen(string);

for (i = 0; i < len; i++) {

glutBitmapCharacter(GLUT\_BITMAP\_HELVETICA\_18, string[i]);

}

glLineWidth(5.0f);

glBegin(GL\_LINE\_LOOP);

glVertex2i(5,5);

glVertex2i(5,475);

glVertex2i(635,475);

glVertex2i(635,5);

glEnd();

int len1;

glRasterPos2f(140,420);

char string1[]="By Abhishek Kumar & Aditya Krishnan";

len1 = (int)strlen(string1);

for (i = 0; i < len1; i++) {

glutBitmapCharacter(GLUT\_BITMAP\_HELVETICA\_18, string1[i]);

}

glLineWidth(5.0f);

glBegin(GL\_LINE\_LOOP);

glVertex2i(100,100);

glVertex2i(100,350);

glVertex2i(550,350);

glVertex2i(550,100);

glEnd();

glColor3f( 0.0f, 0.0f, 1.0f );

glBegin(GL\_POLYGON);

glVertex2i(100,100);

glVertex2i(100,350);

glVertex2i(550,350);

glVertex2i(550,100);

glEnd();

glColor3f( 1.0f, 1.0f, 0.0f );

if (counter >= 1)

{

glColor3f( 1.0f, 1.0f, 0.0f );

glBegin(GL\_POLYGON);

glVertex2i(110,110);

glVertex2i(110,140);

glVertex2i(540,140);

glVertex2i(540,110);

glEnd();

glColor3f( 1.0f, 0.0f, 0.0f );

int len1;

glRasterPos2f(320,120);

char string1[]="1";

len1 = (int)strlen(string1);

for (i = 0; i < len1; i++) {

glutBitmapCharacter(GLUT\_BITMAP\_HELVETICA\_18, string1[i]);

}

}

if (counter >= 2)

{

glColor3f( 1.0f, 1.0f, 0.0f );

glBegin(GL\_POLYGON);

glVertex2i(110,160);

glVertex2i(110,190);

glVertex2i(540,190);

glVertex2i(540,160);

glEnd();

glColor3f( 1.0f, 0.0f, 0.0f );

int len1;

glRasterPos2f(320,170);

char string1[]="2";

len1 = (int)strlen(string1);

for (i = 0; i < len1; i++) {

glutBitmapCharacter(GLUT\_BITMAP\_HELVETICA\_18, string1[i]);

}

}

if (counter >= 3)

{

glColor3f( 1.0f, 1.0f, 0.0f );

glBegin(GL\_POLYGON);

glVertex2i(110,210);

glVertex2i(110,240);

glVertex2i(540,240);

glVertex2i(540,210);

glEnd();

glColor3f( 1.0f, 0.0f, 0.0f );

int len1;

glRasterPos2f(320,220);

char string1[]="3";

len1 = (int)strlen(string1);

for (i = 0; i < len1; i++) {

glutBitmapCharacter(GLUT\_BITMAP\_HELVETICA\_18, string1[i]);

}

}

if (counter >= 4)

{

glColor3f( 1.0f, 1.0f, 0.0f );

glBegin(GL\_POLYGON);

glVertex2i(110,260);

glVertex2i(110,290);

glVertex2i(540,290);

glVertex2i(540,260);

glEnd();

glColor3f( 1.0f, 0.0f, 0.0f );

int len1;

glRasterPos2f(320,270);

char string1[]="4";

len1 = (int)strlen(string1);

for (i = 0; i < len1; i++) {

glutBitmapCharacter(GLUT\_BITMAP\_HELVETICA\_18, string1[i]);

}

}

if (counter >= 5)

{

glColor3f( 1.0f, 1.0f, 0.0f );

glBegin(GL\_POLYGON);

glVertex2i(110,310);

glVertex2i(110,340);

glVertex2i(540,340);

glVertex2i(540,310);

glEnd();

glColor3f( 1.0f, 0.0f, 0.0f );

int len1;

glRasterPos2f(320,320);

char string1[]="5";

len1 = (int)strlen(string1);

for (i = 0; i < len1; i++) {

glutBitmapCharacter(GLUT\_BITMAP\_HELVETICA\_18, string1[i]);

}

}

glColor3f( 1.0f, 1.0f, 1.0f );

if(counter > 5){

if(counter>6)

counter = 6;

glRasterPos2f(250, 50);

int len, i;

char string[]="Stack is full";

len = (int)strlen(string);

for (i = 0; i < len; i++) {

glutBitmapCharacter(GLUT\_BITMAP\_HELVETICA\_18, string[i]);

}

}

if(counter < 0){

if(counter<-1)

counter = -1;

glRasterPos2f(250, 50);

int len, i;

char string[]="Stack is empty";

len = (int)strlen(string);

for (i = 0; i < len; i++) {

glutBitmapCharacter(GLUT\_BITMAP\_HELVETICA\_18, string[i]);

}

}

glFlush();

}

void keys(unsigned char key,int x,int y)

{

if(key=='p')

counter++;

if(key=='P')

counter++;

if(key=='o' || key=='O')

counter--;

}

void main(int argc, char\*\* argv)

{

glutInit(&argc,argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowSize(500,500);

glutInitWindowPosition(100,150);

glutCreateWindow("Stack Implementation Using Glut");

glutDisplayFunc(display);

glutIdleFunc(display);

glutKeyboardFunc(keys);

myInit();

glutMainLoop();

}