NAME : Vivian Vijay Ludrick
ROLL NO.: 9914
BRANCH: SE Comps-A Batch-B

1. WAP for static implementation of stack of floats.

```c
// Stack of floats using arrays
#include <stdio.h>
#include <stdlib.h>

#define SIZE 10 //

// Structure for a stack of array of floats
typedef struct
{
  float arr[SIZE]; // array of floats
  int tos;         // the index of the last elements of the stack
} Stack;

/*
pushes an element to the top of the stack
inputs the Stack as a pointer:have direct access to the stack.
inputs the element: to push on tos
*/
void push(Stack *stk, float ele)
{
  // -> used to access the element through structure's pointer
  if (stk->tos == SIZE - 1)
  {
    printf("Stack Overflow\n\n");
  }
  else
  {
    // increment the tos by one
    stk->tos++;
    stk->arr[stk->tos] = ele; // push at tos
    printf("The pushed element is %f\nCurrently the tos is %d\n\n", ele,
stk->tos);
  }
}
```

```c
/*
peeks the last element of the stack. and decrements the tos.
if we push an element then the element will directly get replaced so we
don't have to worry about deleting the element.
hence only the stack is referenced through the address since we don't need
the element
*/
float pop(Stack *stk)
{
  // checks whether the stack is empty or not
  if (stk->tos == -1)
  {
    printf("Stack Underflow\n\n");
    return (0);
  }
  else
  {
    printf("The popped element is %f\nNow the tos is %d\n\n",
stk->arr[stk->tos], stk->tos - 1);
    return (stk->arr[stk->tos--]);
  }
}

/*
returns the element at the top of the stack
stack is not referenced by address since we only want to view the value and
not edit it
*/
float peek(Stack stk)
{
  return stk.arr[stk.tos]; // return top element of stack
}

/*
Displays all the elements of the stack
stack is not referenced by address since we only want to view the value and
not edit it
*/
void display(Stack stk)
{
```

```
  int i;
  printf("Elements of the stack are:\n\n");
  for (i = stk.tos; i >= 0; i--)
  {
    printf("\t|%f|\n", stk.arr[i]);
  }
}

// Main starts
int main()
{
  Stack s1; // initializing the stack
  float element, dataAtTos;
  int option;
  s1.tos = -1; // top of stack initialized to -1
  // while true
  while (1)
  {
    printf("Enter option of the choice 1.Push 2.Pop 3.Peek 4.Display
5.Close\n");
    scanf("%d", &option);
    switch (option)
    {
    case 1:
      printf("Enter the element to push:\t\n");
      scanf("%f", &element);
      push(&s1, element);
      break;
    case 2:
      dataAtTos = pop(&s1);
      break;
    case 3: // peeks the top of stack
      printf("The element at tos is :\t%f\n\n", peek(s1));
      break;
    case 4:
      display(s1);
      break;
    case 5:
      exit(0); // exit from program with exit status 0
    }
```

```
  }
  return 0;
}
```

OUTPUT:

```
Enter option of the choice 1.Push 2.Pop 3.Peek 4.Display 5.Close
1
Enter the element to push:
4
The pushed element is 4.000000
Currently the tos is 0

Enter option of the choice 1.Push 2.Pop 3.Peek 4.Display 5.Close
1
Enter the element to push:
70
The pushed element is 70.000000
Currently the tos is 1

Enter option of the choice 1.Push 2.Pop 3.Peek 4.Display 5.Close
4
Elements of the stack are:

        |70.000000|
        |4.000000|
Enter option of the choice 1.Push 2.Pop 3.Peek 4.Display 5.Close
3
The element at tos is : 70.000000

Enter option of the choice 1.Push 2.Pop 3.Peek 4.Display 5.Close
2
The popped element is 70.000000
Now the tos is 0

Enter option of the choice 1.Push 2.Pop 3.Peek 4.Display 5.Close
4
Elements of the stack are:

        |4.000000|
Enter option of the choice 1.Push 2.Pop 3.Peek 4.Display 5.Close
5
```

POSTLAB:

WAP for static implementation of stack of books.

```c
// Stack of floats using arrays
#include <stdio.h>
#include <stdlib.h>

#define SIZE 10        // size of stack
#define STRING_SIZE 50 // max size of strings

// Structure of book
typedef struct
{
  int id;
  char title[STRING_SIZE];
  char author[STRING_SIZE];
  int price;
} Book;

// Structure for a stack of array of floats
typedef struct
{
  Book books[SIZE]; // array of Book struct
  int tos;          // the index of the last elements of the stack
} Stack;

/*
pushes an element to the top of the stack
inputs the Stack as a pointer:have direct access to the stack.
inputs the element: to push on tos
*/
void push(Stack *stk, Book curBook)
{
  // -> used to access the element through structure's pointer
  // Note:pre-increment required.
  stk->tos++;
  stk->books[stk->tos] = curBook; // push at tos
  printf("\nThe pushed book is:\n\tBook Name :\t%s\n\tBook Author
:\t%s\n\tBook Id :\t%d\n\tBook Price :\tRs.%d/-\nCurrently the tos is
:\t%d\n\n", curBook.title, curBook.author, curBook.id, curBook.price,
stk->tos);
```

```
}

/*
peeks the last book of the stack. and decrements the tos.
if we push an element then the element will directly get replaced so we
don't have to worry about deleting the element.
hence only the stack is referenced through the address since we don't need
the element
*/
Book pop(Stack *stk)
{
  // checks whether the stack is empty or not
  if (stk->tos == -1)
  {
    printf("Stack Underflow\n\n");
    Book nullBook = {.id = 0, .title = "", .author = "", .price = 0};
    return nullBook;
  }
  else
  {
    // expansion of stk->books[stk->tos]title is
(*stk).(*(books+(*stk).tos)).name; here name doesn't require an arrow
operator because books are not in address format.
    printf("\nThe popped book is:\n\tBook Name :\t%s\n\tBook Author
:\t%s\n\tBook Id :\t%d\n\tBook Price :\tRs.%d/-\nCurrently the tos is
:\t%d\n\n", stk->books[stk->tos].title, stk->books[stk->tos].author,
stk->books[stk->tos].id, stk->books[stk->tos].price, stk->tos - 1);
    return (stk->books[stk->tos--]);
  }
}

/*
returns the book at the top of the stack
stack is not referenced by address since we only want to view the value and
not edit it
*/
Book peek(Stack stk)
{
  return stk.books[stk.tos]; // return top element of stack
}
```

```c
/*
Displays all the books of the stack
stack is not referenced by address since we only want to view the value and
not edit it
*/
void display(Stack stk)
{
  int i;
  printf("\nThe stack of books is:\n\n");
  for (i = stk.tos; i >= 0; i--)
  {
    printf("\tBook Name :\t%s\n\tBook Author :\t%s\n\tBook Id :\t%d\n\tBook
Price :\tRs.%d/-\n", stk.books[i].title, stk.books[i].author,
stk.books[i].id, stk.books[i].price);
  }
  printf("\n");
}

int main()
{
  Stack s1;       // initializing the stack
  Book curBook;   // the book to be passed at various functions is stored
here
  Book dataAtTos; // last book of the stack
  int option;     // selection of operation on the stack
  s1.tos = -1;    // top of stack initialized to -1. If not done then you
get segmentation fault

  // while true
  while (1)
  {
    printf("Enter the option of the choice 1.Push 2.Pop 3.Peek 4.Display
5.Close\n");
    scanf("%d", &option);

    switch (option)
    {
    case 1:
```

```
      // defined here so that the user doesn't have to enter all the details
and then see that the stack has overflown.
      if (s1.tos == SIZE - 1)
      {
        printf("Stack Overflow\n\n");
      }
      else
      {
        printf("\nEnter the name of the book:\t");
        fgets(curBook.title, SIZE, stdin);
        scanf("%[^\n]", &curBook.title);
        printf("Enter the author of the book:\t");
        fgets(curBook.author, SIZE, stdin);
        scanf("%[^\n]", &curBook.author);
        printf("Enter the id of the book:\t");
        scanf("%d", &curBook.id);
        printf("Enter the price of the book in Rs:\t");
        scanf("%d", &curBook.price);
        push(&s1, curBook); // pushes the book on the stack
        break;
      }
    case 2:
      dataAtTos = pop(&s1); // stores the return value of pop for future use
      break;
    case 3: // peeks the top of stack
      dataAtTos = peek(s1);
      printf("The book at tos is:\n\tBook Name :\t%s\n\tBook Author
:\t%s\n\tBook Id :\t%d\n\tBook Price :\tRs.%d/-\nCurrently the tos is
:\t%d\n\n", dataAtTos.title, dataAtTos.author, dataAtTos.id,
dataAtTos.price, s1.tos);
      break;
    case 4:
      display(s1);// displays all elements of the stack
      break;
    case 5:
      exit(0); // exit from program with exit status 0
    default:
      printf("Enter a valid option\n");
    }
  }
```

```
  return 0;
}
```

OUTPUT:

```
Enter the option of the choice 1.Push 2.Pop 3.Peek 4.Display 5.Close
1

Enter the name of the book:     Art of War
Enter the author of the book:   Sun Tzu
Enter the id of the book:       1
Enter the price of the book in Rs:      200

The pushed book is:
        Book Name :     Art of War
        Book Author :   Sun Tzu
        Book Id :       1
        Book Price :    Rs.200/-
Currently the tos is :  0

Enter the option of the choice 1.Push 2.Pop 3.Peek 4.Display 5.Close
1

Enter the name of the book:     Rich Dad Poor Dad
Enter the author of the book:   Robert Kiyosaki
Enter the id of the book:       2
Enter the price of the book in Rs:      335

The pushed book is:
        Book Name :     Rich Dad Poor Dad
        Book Author :   Robert Kiyosaki
        Book Id :       2
        Book Price :    Rs.335/-
Currently the tos is :  1

Enter the option of the choice 1.Push 2.Pop 3.Peek 4.Display 5.Close
3
The book at tos is:
        Book Name :     Rich Dad Poor Dad
        Book Author :   Robert Kiyosaki
        Book Id :       2
        Book Price :    Rs.335/-
Currently the tos is :  1
```

```
Enter the option of the choice 1.Push 2.Pop 3.Peek 4.Display 5.Close
4

The stack of books is:

        Book Name :     Rich Dad Poor Dad
        Book Author :   Robert Kiyosaki
        Book Id :       2
        Book Price :    Rs.335/-
        Book Name :     Art of War
        Book Author :   Sun Tzu
        Book Id :       1
        Book Price :    Rs.200/-

Enter the option of the choice 1.Push 2.Pop 3.Peek 4.Display 5.Close
2

The popped book is:
        Book Name :     Rich Dad Poor Dad
        Book Author :   Robert Kiyosaki
        Book Id :       2
        Book Price :    Rs.335/-
.Currently the tos is : 0

Enter the option of the choice 1.Push 2.Pop 3.Peek 4.Display 5.Close
4

The stack of books is:

        Book Name :     Art of War
        Book Author :   Sun Tzu
        Book Id :       1
        Book Price :    Rs.200/-

Enter the option of the choice 1.Push 2.Pop 3.Peek 4.Display 5.Close
5
```