DAA Assignment no. 1 – B

Name : Aditya Rajesh Jadhav.

Reg. No.: 2020BIT044

*stack :

```
#include<stdio.h>
#include<stdlib.h>

#define max 10;
int stack_arr[max];
int top = -1;
int isfull(){
   if(top == (max - 1)){
      return 1;
   }
   else{
      return 0;
   }
}
int isempty(){
   if(top == -1){
      return 1;
   }
   else{
      return 0;
```

```
    }
}
void push(int item){
    if(isfull()){
        printf("Stack overflow.\n");
        return;
    }
    top = top + 1;
    stack_arr[top] = item;
}

int pop(){
    if(isempty()){
        printf("Stack underflow.\n");
        return;
    }
    item = stack_arr[top];
    top = top - 1;

    return item;
}

int peak(){
    if(isempty()){
        printf("Stack underflow.\n");
        exit;
```

```
    }
    return stack_arr[top];
}

void display(){
    if(isempty()){
        printf("Stack underflow.\n");
        exit;
    }
    for(int i = top;i>=0; i--){
        printf("%d\n",i);
    }
}

void main()
{
    /* code */
    int item, choice;
    printf("1. Push \n");
    printf("2. Pop\n ");
    printf("3. Peak \n");
    printf("4. Display\n ");
    printf("Enter your choice - \n");
    scanf("%d",&choice);
    switch (choice)
    {
```

```
    case 1:
        printf("Enter the item that you have to push into your stack -\n");
        scanf("%d",&item);
        push(item);
        break;
    case 2:
        printf("You can only pop out the last element form your stack .\n");
        item = pop();
        printf("So your poped element is - %d \n",item) ;
        break;
    case 3:
        item = peak();
        printf("Item at the top of your stack is - %d \n" ,item) ;
        break;
    case 4:
        display ();
        break;

    default:
        break;
    }

}
```

*Queue:

```
#include <iostream>
```

```
using namespace std;
int queue[100], n = 100, front = - 1, rear = - 1;
void Insert() {
  int val;
  if (rear == n - 1)
  cout<<"Queue Overflow"<<endl;
  else {
    if (front == - 1)
    front = 0;
    cout<<"Insert the element in queue : "<<endl;
    cin>>val;
    rear++;
    queue[rear] = val;
  }
}
void Delete() {
  if (front == - 1 || front > rear) {
    cout<<"Queue Underflow ";
    return ;
  } else {
    cout<<"Element deleted from queue is : "<< queue[front] <<endl;
    front++;;
  }
}
void Display() {
  if (front == - 1)
```

```
  cout<<"Queue is empty"<<endl;
  else {
    cout<<"Queue elements are : ";
    for (int i = front; i <= rear; i++)
    cout<<queue[i]<<" ";
      cout<<endl;
  }
}
int main() {
  int ch;
  cout<<"1) Insert element to queue"<<endl;
  cout<<"2) Delete element from queue"<<endl;
  cout<<"3) Display all the elements of queue"<<endl;
  cout<<"4) Exit"<<endl;
  do {
    cout<<"Enter your choice : "<<endl;
    cin>>ch;
    switch (ch) {
      case 1: Insert();
      break;
      case 2: Delete();
      break;
      case 3: Display();
      break;
      case 4: cout<<"Exit"<<endl;
      break;
```

```
        default: cout<<"Invalid choice"<<endl;
      }
   } while(ch!=4);
   return 0;
}
```