Dr. D.Y.Patil Arts, Commerce, Science College,

Pimpri, Pune-18

S.Y.B.Sc(Comp. Sci) 2024-25

Data Structures and Algorithms – I

Assignment 7: Stack

Set A

a) Implement a stack library (ststack.h) of integers using a static implementation of the stack and implementing the above six operations. Write a driver program that includes stack library and calls different stack operations.

```
#define MAX 5
struct STACK
{
   int stk[MAX];
   int top;
};
typedef struct STACK stack;

//initialize the stack
void initstack(stack *s)
{
   int i;
   for(i=0;i<MAX;i++)
    s->stk[i]=0;
   s->top=-1;
}
int isempty(stack *s)
{
    if(s->top==-1)
      return 1;
    else
      return 0;
}
int isfull(stack *s)
{
   if(s->top==MAX-1)
     return 1;
   else
     return 0;
}
void push(stack *s,int data)
{
   s->top++;
   s->stk[s->top]=data;

}
int pop(stack *s)
{
   int val;
   val=s->stk[s->top];
   s->top--;
   return(val);  //return(s->stk[s->top--]);
}
```

```
int peek(stack *s)
{
   return(s->stk[s->top]);
}

void display(stack *s)
{
   int i;
   for(i=0;i<=s->top;i++)
      printf("\t%d",s->stk[i]);
}

#include<stdio.h>
#include"stack.h"
main()
{
    stack s;
    int ch,data;
    initstack(&s);
    while(1)
    {
    printf("\nMain Menu.");
    printf("\n1:PUSH.");
    printf("\n2:POP.");
    printf("\n3:PEEK.");
    printf("\n4:Display.");
    printf("\n5:Exit.");
    printf("\nEnter the Choice:");
    scanf("%d",&ch);
    switch(ch)
    {
      case 1:   if(isfull(&s))
                 printf("\nStack is FULL.");
           else
           {
           printf("\nEnter the data to PUSH:");
           scanf("%d",&data);
           push(&s,data);
           }
           break;
      case 2:   if(isempty(&s))
                printf("\nStack is empty.");
           else
              printf("\nPopped data from stack is %d",pop(&s));
           break;
     case 3:   if(isempty(&s))
                printf("\nStack is empty.");
           else
              printf("\nTop data from stack is %d",peep(&s));
           break;

      case 4: if(isempty(&s))
                printf("\nStack is empty.");
               else
                display(&s);
           break;
      case 5: exit(0);
     }
    }
}
```

b) Implement a stack library (dystack.h) of integers using a dynamic (linked list) implementation of the stack and implementing the above five operations. Write a driver program that includes stack library and calls different stack operations.

```
#include<stdio.h>
#include"stackdy.h"
main()
{
  int ch,data;
  while(1)
  {
  printf("\n1:PUSH.");
  printf("\n2:POP.");
  printf("\n3:Peek.");
  printf("\n4:Display.");
  printf("\n5:Exit.");
  printf("\nEnter the Choice:");
  scanf("%d",&ch);
  switch(ch)
  {
     case 1: printf("\nEnter the data:");
             scanf("%d",&data);
             push(data);
             break;

     case 2:  if(isempty())
                printf("\nstack is empty.");
             else
                printf("\nPopped data from stack: %d",pop());
             break;
     case 3:  if(isempty())
                printf("\nstack is empty.");
             else
                printf("\nTop data from stack: %d",peek());
             break;

     case 4:    if(isempty())
                 printf("\nStack is Empty.");
             else
                 display();
             break;
     case 5:exit(0);
  }
  }
}

struct NODE
{
   int data;
   struct NODE *next;
};
typedef struct NODE node;
node *top=NULL;
node *getnodenum(int data)
{
   node *temp;
   temp=(node*)malloc(sizeof(node));
   temp->data=data;
   temp->next=NULL;
   return(temp);
}
int isempty()
{
   if(top==NULL)
     return 1;
   else
     return 0;
}
void push(int data)
{
   node *temp;
   temp=getnodenum(data);
   temp->next=top;
   top=temp;
}
int pop()
{
  int val;
  node *ptr;
  ptr=top;
  val=ptr->data;
  top=ptr->next;
  free(ptr);
  return val;
}
int peek()
{
  return top->data;
}

void display()
{
   node *ptr;
   for(ptr=top;ptr!=NULL;ptr=ptr->next)
      printf("\t%d",ptr->data);
}
```

Set B

a) Write a program to check whether the contents of two stacks are identical. Use stack library to perform basic stack operations. Neither stack should be changed.

b) Write a program that copies the contents of one stack into another. Use stack library to perform basic stack operations. The order of two stacks must be identical.(Hint: Use a temporary stack to preserve the order).