C- Program to implement STACK using arrays / global variables.

(1)

```
# include <stdio.h>
# include <process.h>
# include <conio.h>
# define STACK_SIZE 5
int top = -1;
int s[10];
int item;
void push()
{
  if(top == STACK_SIZE - 1)
  {
    printf("stack overflow \n");
    return;
  }
  top = top + 1;
  s[top] = item;
}
int pop()
{
if (top == -1) return -1;
return s[top--];
}
void display()
{ int i;
  if (top == -1)
  { printf ("stack is empty \n");
    return;
  }
  printf ("contents of the stack \n");
  for (i=0; i <= top; i++)
    { printf ("%d \n", s[i]);
    }
}
```

```
void main()
{ int item_deleted;
  int choice;
  clrscr();
  for(;;)
  {
  printf("\n 1: push \n 2: pop \n 3. display \n 4: exit \n");
  printf("enter the choice \n");
  scanf("%d", &choice);
  switch(choice)
  {
  case 1: printf("enter the item to be inserted \n");
          scanf("%d", &item);
          push();
          break;
  case 2: item_deleted = pop();
          if(item_deleted == -1)
          printf("stack is empty \n");
          else
          printf("item deleted is %d \n", item_deleted);
          break;
  case 3: display();
          break;
  default: exit(0);
  }
  }
  getch();
}
```

C- Program to implement STACK using ~~array~~ pointer

(2)

```
#include <stdio.h>
#include <process.h>
#include <conio.h>
#define STACK_SIZE 5
int top = -1;
void push(int item, int s[], int *top)
{
  if(*top == STACK_SIZE - 1)
  {
    printf("stack overflow\n");
    return;
  }
  *top = *top + 1;
  s[*top] = item;
}
int pop(int s[], int *top)
{
  int item_deleted;
  if (*top == -1)
  {
    printf("stack underflow cannot delete\n");
    return 0;
  }
  item_deleted = s[*top];
  *top = *top - 1;
  return item_deleted;
}
void display(int top, int s[])
{
  int i;
  if (top == -1)
  {
    printf("stack is empty\n");
    return;
  }
  printf("contents of the stack\n");
```

```
for(i=0;i<=top;i++)
{
  printf("%d\n",s[i]);
}
}
void main()
{
  int item,s[10];
  int item_deleted;
  int choice;
  clrscr();
  for(;;)
  {
  printf("\n1: push\n2: pop\n3: display\n4: exit\n");
  printf("enter the choice\n");
  scanf("%d",&choice);
  switch(choice)
  {
  case 1: printf("enter the item to be inserted\n");
          scanf("%d",&item);
          push(item,s,&top);
          if(item_deleted!=0) printf("item deleted is %d\n",item_deleted);
          break;
  case 2: item_deleted = pop(s,&top);
          if(item_deleted!=0)
          printf("item deleted is %d\n",item_deleted);
          break;
  case 3: display(top,s);
          break;
  default: exit(0);
  }
  }
  getch();
}
```