```
#include<stdio.h>
#define size 3
int top=-1;
void push(int[], int);
int pop(int[]);
void display(int[]);

int main(int argc, char *argv)
  {
  int size, int[size];
  int choice, element;
  char ch;
  do
  { printf("Enter size of array\n");
    scanf("%d", &size);
  printf("Enter your choice\n 1. Push\n
  2. Pop\n 3. Display\n");
   scanf("%d", &choice);
    switch(choice)
     {
     case 1:
     printf("Enter the element to be pushed\n");
     scanf("%d", &element);
     push(stack, element);
     break;
     case 2:
     element = pop(stack);
     if(element == -1)
        printf("Stack Underflow");
     else
     printf("Poped element is %d\n",
       element);
     break;
     case 3:
     display(stack);
     break;
```

```
    {
        printf("%d\t", stack[i]);
    }
}
int main(int argc, char *argv)
{
    int size, int [size];
    int choice, element;
    char ch;
    do
    { printf("Enter size of array\n");
      scanf("%d", &size);
     printf("Enter your choice\n 1. Push \n
    2. Pop\n 3. Display \n");
     scanf("%d", &choice);
     switch (choice)
      {
        case 1:
        printf("Enter the element to be pushed\n");
        scanf("%d", &element);
        push (stack, element);
        break;
        case 2:
        element = pop(stack);
        if(element == -1)
            printf("Stack Underflow");
        else
            printf("Poped element is %d\n",
            element);
        break;
       case 3:
        display(stack);
        break;
```

```
void push (int [], int ele)
{
   if (top == size-1)
   {
      printf ("Stack overflow");
   }
   else
   {
      top++;
      stack [top] = ele;
   }
}
int pop (int stack[])
{
   int popele;
   if (top == -1)
      return -1;
   else
   {
      popele = stack [top];
      top--;
      return (popele);
   }
}
void display (int stack [])
{
   int i;
   printf ("The stack elements \n");
   for (i=top; i >=0; i--)
```

```
default:
        printf("Invalid choice");
}

printf("Do you want to continue:\n");
fflush(stdin);
scanf("%c",&ch);
}
while (ch == 'y' || ch == 'Y');
return 0;
}
```

```
    }
    printf("%d\t", stack[i]);
}
```