Data Structure

Write a program to implement following STACK operations.
(1)PUSH (2)POP (3)PEEP (4)CHANGE (5)DISPLAY

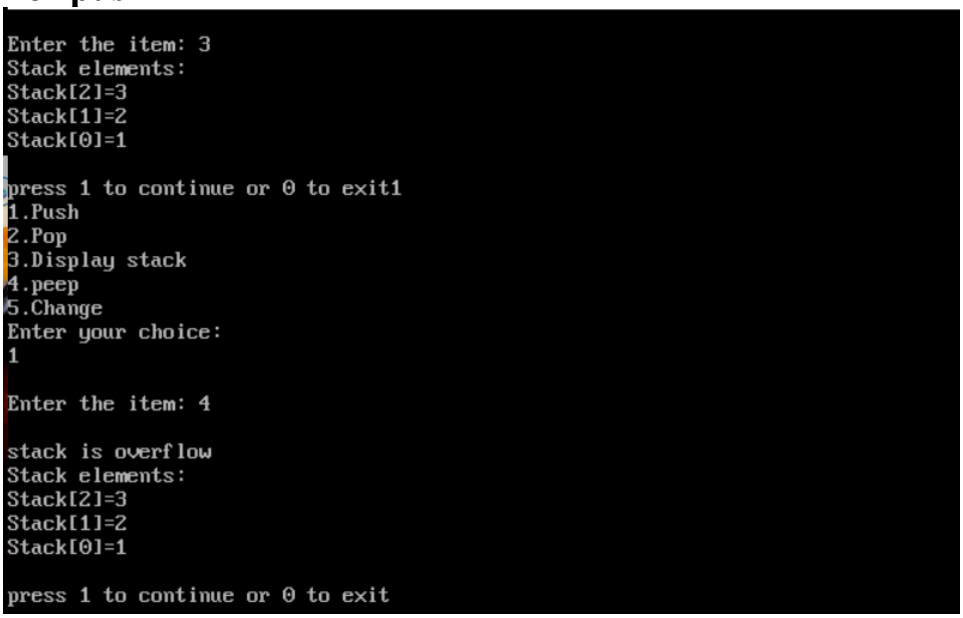
```
#include<stdio.h>
#include<conio.h>
#define MAX 3

int stack[MAX];
int top = -1;

int pop();
void  push(int item);
void display();
void peep();
void change();

void main()
{
     int choice, item, ans = 1;
     clrscr();
     while(ans != 0)
     {
          printf("1.Push\n");
          printf("2.Pop\n");
          printf("3.Display stack\n");
          printf("4.peep\n");
          printf("5.Change\n");
          printf("Enter your choice:\n");
          scanf("%d",&choice);
          switch(choice)
          {
               case 1:
                    printf("\nEnter the item: ");
                    scanf("%d" ,&item);
                    push(item);
                    break;
               case 2:
                    item=pop();
                    printf("popped item is: %d\n",item);
                    break;
               case 3:
                    display();
                    break;
               case 4:
                    peep();
                    break;
               case 5:

                    change();
                    display();
                    break;
               default:
                    printf("\nWrong choice\n");
```

```
            }
            printf("\npress 1 to continue or 0 to exit");
            scanf("%d",&ans);
      }
      getch();
}
void push(int item)
{
      if(top==(MAX-1))
      {
            printf("\nstack is overflow");
            return;
      }
      top=top+1;
      stack[top]=item;
}
int pop()
{
      int item;
      if(top==-1)
      {
            printf("\nstack is underflow");
            exit(1);
      }
      item= stack[top];
      top=top-1;
      return item;
}
void display()
{
       int i;
       if(top==-1)
       {
            printf("\nstack is empty");
            return;
       }
       printf("Stack elements:\n ");
       for(i=top;i>=0;i--)
       {
            printf("Stack[%d]=%d\n",i,stack[i]);
       }
}
void change()
{
      int i,it;
      printf("Enter the location you want to change: ");
      scanf("%d",&i);
      printf("\nEnter new element:");
      scanf("%d",&it);
      if(top == -1)
      {
            printf("\nStack is empty");
            return;
      }
      stack[top-i]=it;
}
```

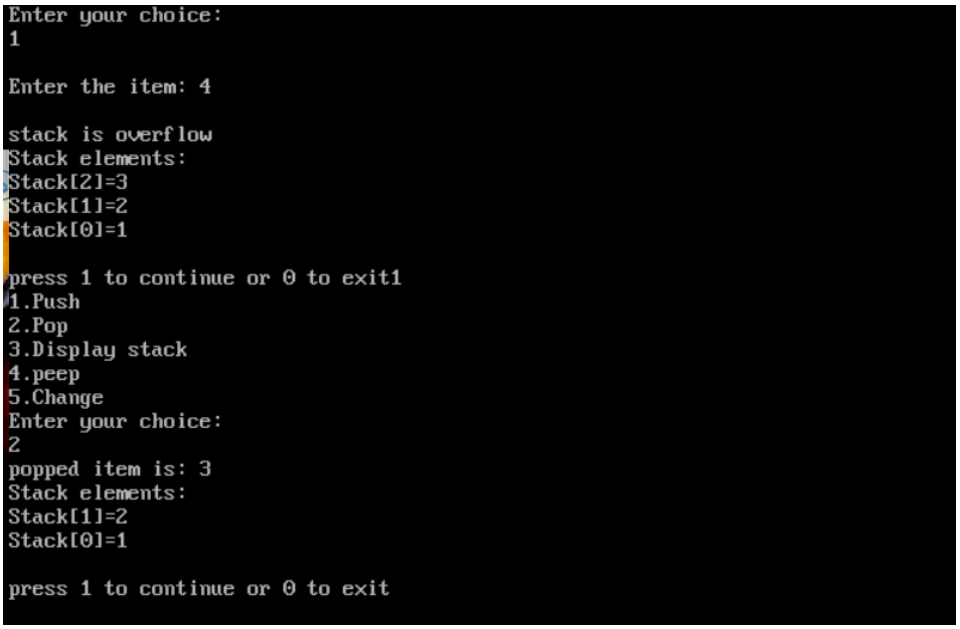
```
void peep()
{
      int i;
      printf("Enter the location you want to peep: ");
      scanf("%d",&i);

      if(top == -1)
      {
            printf("\nStack is empty");
            return;
      }
      printf("\npeeped element is stack[%d] : %d",top-i,stack[top-i]);
}
```

OUTPUT:

For push



For POP



For PEEP

```
2.Pop
3.Display stack
4.peep
5.Change
Enter your choice:
1

Enter the item: 67
Stack elements:
Stack[2]=67
Stack[1]=2
Stack[0]=1

press 1 to continue or 0 to exit1
1.Push
2.Pop
3.Display stack
4.peep
5.Change
Enter your choice:
4
Enter the location you want to peep: 2

peeped element is stack[0] : 1
press 1 to continue or 0 to exit
```

For CHANGE

```
3.Display stack
4.peep
5.Change
Enter your choice:
4
Enter the location you want to peep: 2

peeped element is stack[0] : 1
press 1 to continue or 0 to exit1
1.Push
2.Pop
3.Display stack
4.peep
5.Change
Enter your choice:
5
Enter the location you want to change: 2

Enter new element:88
Stack elements:
Stack[2]=67
Stack[1]=2
Stack[0]=88

press 1 to continue or 0 to exit
```

For DISPLAY

```
5.Change
Enter your choice:
5
Enter the location you want to change: 2

Enter new element:88
Stack elements:
Stack[2]=67
Stack[1]=2
Stack[0]=88

press 1 to continue or 0 to exit1
1.Push
2.Pop
3.Display stack
4.peep
5.Change
Enter your choice:
3
Stack elements:
Stack[2]=67
Stack[1]=2
Stack[0]=88

press 1 to continue or 0 to exit
```