stack_out.c

```c
#include<stdio.h>
#define maxsize 5
int stack [maxsize];
int top=-1;

void push (int item){
    if (top>=maxsize-1){
        printf("The stack is full\n");
    }

    else
    {
        top++;
        stack[top]=item;
    }
}

int pop()
{
    if (top<0)
    {
        printf("the stack is empty\n");
        return 0;
    }

    else
    {
        int item=stack[top];
        top=top-1;
        return item;
    }
}

void display()
{
    if (top<0)
    {
        printf("The stack is empty\n");
    }


    else{
        for (int i=0; i<=top; i++)
        {
            printf("The elements are : %d \n",stack[i]);
        }
    }
}
```

```c

int main(){
    int choice, item;

    while(1)
    {
        printf("\n");
        printf("1. Push\n2. Pop\n3. Display\n4. Exit\n");

        printf("Enter choice : ");
        scanf("%d",&choice);

        switch (choice)
        {
            case 1: printf("Choice selected : \"PUSH\" \n");
                    printf("Enter an item to insert : ");
                    scanf("%d",&item);
                    push(item);
                    break;
            case 2: printf("Choice selected : \"POP\" \n");
                    printf("Enter item to be removed : %d\n",pop());
                    break;
            case 3: printf("Choice selected : \"DISPLAY\" \n");
                    display();
                    break;
            case 4: return 0;
            default: printf("Invalid Choice\n");
        }
    }
}
```