Foundation University School of Science and Technology

Data Structure

Lab Manual

Submitted by: WARDA JAVED

094

Submitted to:Sir Sharjeel

LAB 02 TASK

1. Write a program to take numbers from user and stored into stack. Pop the values from

stack and show only even values on screen and show highest marks on the screen.

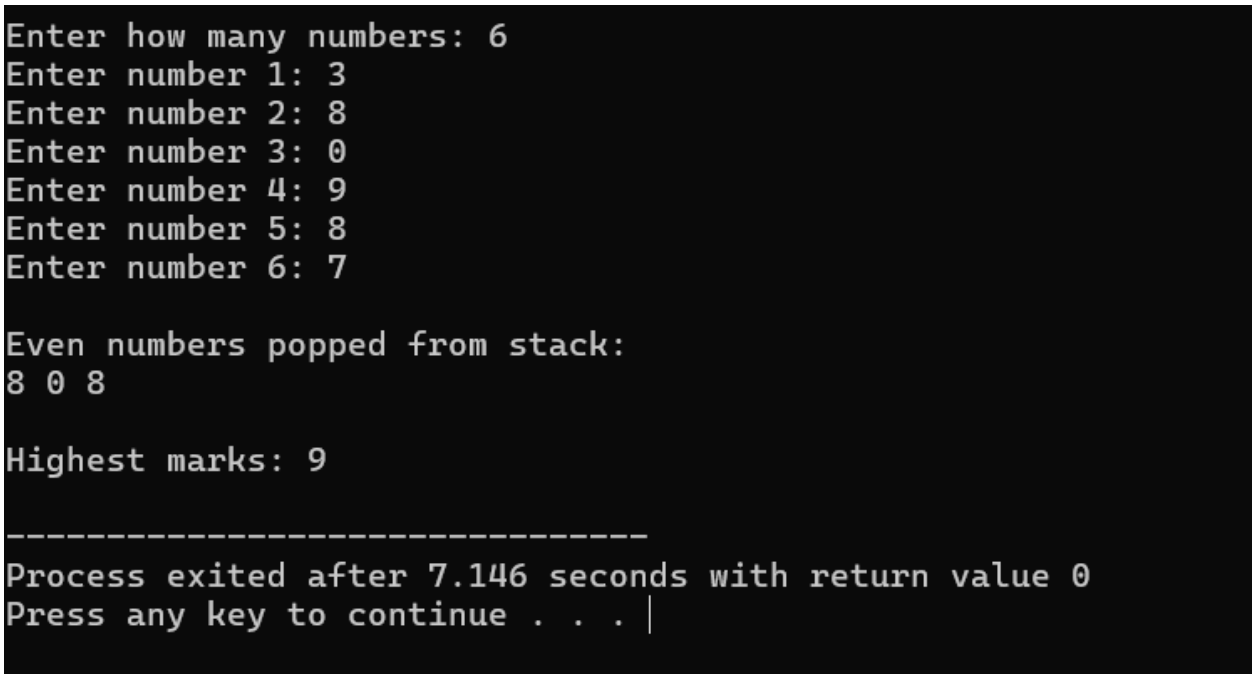
```
#include <stdio.h>

#define SIZE 100

int main() {
    int stack[SIZE], top = -1;
    int n, i, num, highest;
    printf("Enter how many numbers: ");
    scanf("%d", &n);
    // push numbers
    for(i = 0; i < n; i++) {
        printf("Enter number %d: ", i+1);
        scanf("%d", &num);
        stack[++top] = num;
    }
    highest = stack[0];
    printf("\nEven numbers popped from stack:\n");
    while(top >= 0) {
        int val = stack[top--];   // pop
        if(val % 2 == 0) {
            printf("%d ", val);
        }
        if(val > highest) {
            highest = val;
```

```
        }
    }
    printf("\n\nHighest marks: %d\n", highest);
    return 0;
}
```

OUTPUT:



2. Program: Infix to Postfix (Using Stack):

```
#include <stdio.h>

#include <ctype.h>

#define SIZE 100

char stack[SIZE];

int top = -1;

void push(char c) {

  stack[++top] = c;

}
```

```
char pop() {
   return stack[top--];
}
int precedence(char op) {
   if(op == '^') return 3;
   if(op == '*' || op == '/') return 2;
   if(op == '+' || op == '-') return 1;
   return 0;
}
int main() {
   char infix[SIZE], postfix[SIZE];
   int i, j = 0;
   char ch;
   printf("Enter infix expression: ");
   scanf("%s", infix);
   for(i = 0; infix[i] != '\0'; i++) {
      ch = infix[i];
      if(isalnum(ch)) {          // operand
         postfix[j++] = ch;
      }
      else if(ch == '(') {
         push(ch);
      }
      else if(ch == ')') {
         while(top != -1 && stack[top] != '(') {
            postfix[j++] = pop();
```

```
        }
        pop(); // remove '('
      }
      else { // operator
        while(top != -1 && precedence(stack[top]) >= precedence(ch)) {
          postfix[j++] = pop();
        }
        push(ch);
      }
    }
    while(top != -1) {
      postfix[j++] = pop();
    }
    postfix[j] = '\0';
    printf("Postfix expression: %s\n", postfix);
    return 0;
}
```

OUTPUT: