LAB – 3

I.SOLVED EXERCISE:

1)Write a c program to check if the given parenthesized expression has properly matching open and closing parenthesis.

CODE :

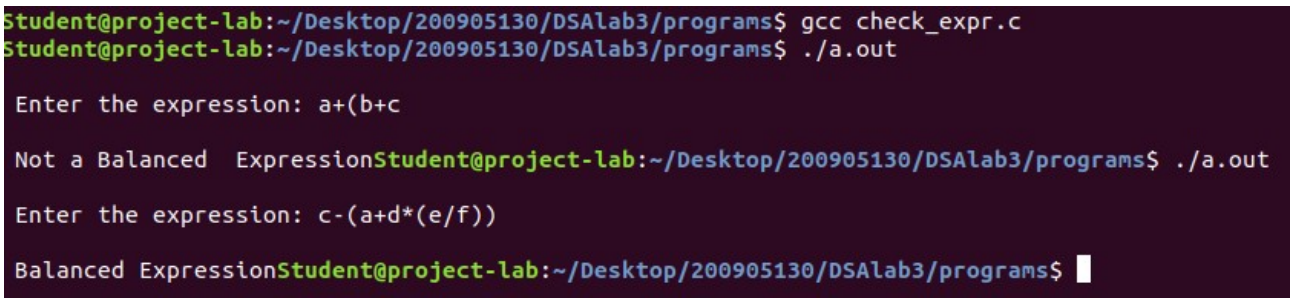
File name: stack_operations.h

```
# define MAX 10
# define true 1
# define false 0
/* Structure definition */
typedef struct
{
 char item[MAX];
 int top;
}stack;
void push(stack *ps,char x);
char pop(stack *ps);
int empty(stack *ps);
/* Push operation */
void push(stack *ps,char x)
{
 if (ps->top!=MAX-1)
 {
  ps->top++;
  ps->item[ps->top]=x;
 }
}



 /* Pop operation */
char pop(stack *ps)
{
 if(!empty(ps))
    return(ps->item[ps->top--]);
}

/* Stack empty operation */
int empty(stack *ps)
{
 if (ps->top==-1)
  return(true);
 else
  return(false);
}
```

File name: check_expr.c

```
#include <stdio.h>
#include <stdlib.h>
#include "stack_operations.h"
void main()
{
 char expn[25],c,d;
 int i=0;
 stack s;
 s.top=-1;
 printf("\n Enter the expression: ");
 scanf("%[^\n]s",expn);
 while((c=expn[i++])!='\0')
 {
  if(c=='(')
   push(&s,c);
  else
   if(c==')')
       {
    d=pop(&s);
    if(d!='(')
    {
    printf("\n Invalid Expression");
                                    break;
    }
   }
 }
 if(empty(&s))
  printf("\n Balanced Expression");
 else
  printf("\n Not a Balanced  Expression");
}
```

OUTPUT :



Questions for Lab3 :

Write a ‘C’ program to:
1) Implement a menu driven program to define a stack of characters. Include push, pop and
display functions. Also include functions for checking error conditions such as underflow
and overflow (ref. figure 1) by defining isEmpty and isFull functions. Use these function
in push, pop and display functions appropriately. Use type defined structure to define a
STACK containing a character array and an integer top. Do not use global variables.

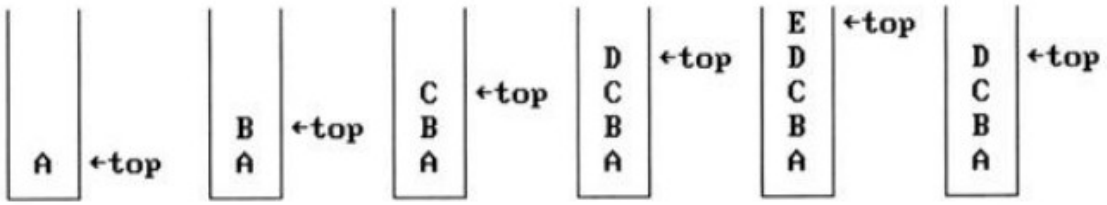


Figure 1: Inserting and deleting elements in a stack

CODE :

Header file : operation.h

```
#define MAX 4
typedef struct
{
   char data[MAX];
   int top;
} stack;
bool isEmpty(stack* s)
{
   if(s->top == -1)
   {
      return true;
   }
   else return false;
}
bool isFull(stack* s)
{
   if(s->top == MAX-1)
   {
      return true;
   }
   else return false;
```

```
}
void push(stack* s, char c)
{
   if(isFull(s))
   {
      printf("Stack is full.\n");
      return;
   }
   s->top++;
   s->data[s->top] = c;
}
char pop(stack* s)
{
   if(!isEmpty(s))
   {
      return(s->data[s->top--]);
   }
}
void display(stack* s)
{
   if(isEmpty(s))
   {
      printf("Stack is empty\n");
      return;
   }
   int count = s->top;
   while(count > -1)
   {
      printf("%c\n",s->data[count--]);
   }
}
```

Program file :

```
#include<stdio.h>
#include<stdbool.h>
#include “operation.h”

int main()
{
   printf("Name : Manoj M Mallya\nRegistraion number : 200905130\nBatch : C2");
   stack st;
   stack* s = &st;
   s->top = -1;
   int n=0;
   char ch;
   do
   {
      printf("\nEnter your choice :\t1 to push\t2 to pop\t3 to display\t4 to exit. : ");
      scanf("%d",&n);
      switch(n)
      {
      case 1 :
```

```
            printf("Enter char to push : ");
            scanf(" %c",&ch);
            push(s,ch);
            break;
        case 2 :
            pop(s);
            break;
        case 3 :
            display(s);
            break;
        case 4 :
            break;
        }
    }

    while(n != 4);
}
```

OUTPUT :

```
Student@project-lab:~/Desktop/200905130/DSAlab3/programs$ gcc lab3_1.c -o stacks
Student@project-lab:~/Desktop/200905130/DSAlab3/programs$ ./stacks
Name : Manoj M Mallya
Registraion number : 200905130
Batch : C2
Enter your choice :     1 to push       2 to pop        3 to display    4 to exit. : 1
Enter char to push : A

Enter your choice :     1 to push       2 to pop        3 to display    4 to exit. : 1
Enter char to push : B

Enter your choice :     1 to push       2 to pop        3 to display    4 to exit. : 1
Enter char to push : C

Enter your choice :     1 to push       2 to pop        3 to display    4 to exit. : 3
C
B
A

Enter your choice :     1 to push       2 to pop        3 to display    4 to exit. : 1
Enter char to push : D

Enter your choice :     1 to push       2 to pop        3 to display    4 to exit. : 3
D
C
B
A

Enter your choice :     1 to push       2 to pop        3 to display    4 to exit. : 2

Enter your choice :     1 to push       2 to pop        3 to display    4 to exit. : 2

Enter your choice :     1 to push       2 to pop        3 to display    4 to exit. : 3
B
A

Enter your choice :     1 to push       2 to pop        3 to display    4 to exit. : 4
```

2) Convert a given decimal number to binary using stack.

CODE :

Header file : conversion.h

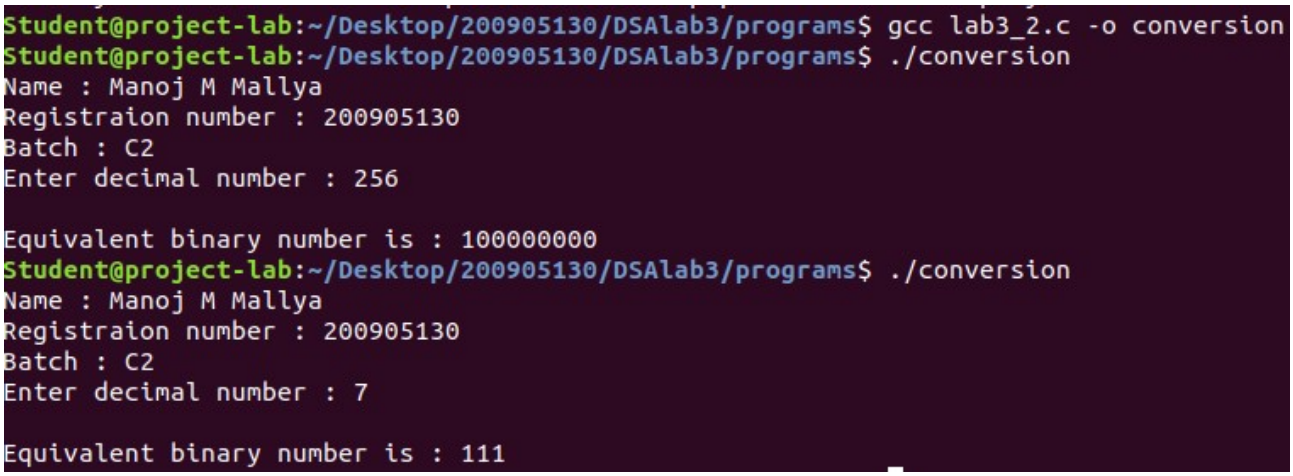
```
#define MAX 10

typedef struct
{
   int data[MAX];
   int top;
} stack;
void push(stack* s, int c)
{
   s->top++;
   s->data[s->top] = c;
}
int pop(stack* s)
{
   return(s->data[s->top--]);
}
void display(stack* s)
{
   int count = s->top;
   while(count > -1)
   {
      printf("%d",s->data[count--]);
   }
}
```

Program :

```
#include <stdio.h>
#include <stdbool.h>
#include ”conversion.h”
int main()
{
   printf("Name : Manoj M Mallya\nRegistraion number : 200905130\nBatch : C2");
   stack st;
   stack* s = &st;
   s->top = -1;
   int n,r;
   printf("\nEnter decimal number : ");
   scanf("%d",&n);
   int q=n;
   while(q!=0)
   {
      r = q%2;
      push(s,r);
      q = q/2;
   }
   printf("\nEquivalent binary number is : ");
```

```
    display(s);
    printf("\n");
}
```

OUTPUT :



3) Determine whether a given string is palindrome or not using stack.

CODE :

Header file : palindrome.h

```
#define MAX 10

typedef struct
{
    char data[MAX];
    int top;
} stack;
void push(stack* s, char e)
{
    s->top++;
    s->data[s->top] = e;
}
char pop(stack* s)
{
    return(s->data[s->top--]);
}
int isPalindrome(stack* s, char str[])
{
    int l = strlen(str);
    int i, m = l/2;
    for(i=0; i<m; i++)
    {
        push(s,str[i]);
    }
    if(l%2!=0)
    {
        i++;
```

```
    }
    char e;
    while(str[i] != '\0')
    {
        e = pop(s);
        if(e!=str[i])
            return 0;
        i++;
    }
    return 1;
}
```

Program :

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "palindrome.h"

int main()
{
    printf("Name : Manoj M Mallya\nRegistraion number : 200905130\nBatch :
C2");
    stack st;
    stack* s = &st;
    s->top = -1;
    char str[20];
    printf("\nEnter the string : ");
    scanf("%s",str);
    if(isPalindrome(s,str))
        printf("\nString is Palindrome.\n");
    else
        printf("\nString is not Palindrome.\n");
    return 0;
}
```

OUTPUT :

```
Student@project-lab:~/Desktop/200905130/DSAlab3/programs$ gcc lab3_3.c -o palindrome
Student@project-lab:~/Desktop/200905130/DSAlab3/programs$ ./palindrome
Name : Manoj M Mallya
Registraion number : 200905130
Batch : C2
Enter the string : racecar

String is Palindrome.
Student@project-lab:~/Desktop/200905130/DSAlab3/programs$ ./palindrome
Name : Manoj M Mallya
Registraion number : 200905130
Batch : C2
Enter the string : technology

String is not Palindrome.
```

****