

DSA LAB

Week 3

Udeet Mittal

CSE C3

Roll Number 64

SOLVED EXERCISE:

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

Filename: “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);
}

```

Filename: “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("Name: Udeet Mittal\nBatch: C3\nRoll Number: 64\n");
    printf("\n Enter the expression: ");
    scanf("%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");
}

```

```
Student@project-lab: ~/Udeet_200905406_C3/Week3
File Edit View Search Terminal Help
Student@project-lab:~/Udeet_200905406_C3/Week3$ gcc check_expr.c
Student@project-lab:~/Udeet_200905406_C3/Week3$ ./a.out
Name: Udeet Mittal
Batch: C3
Roll Number: 64

Enter the expression: a+b+(c/d))

Invalid Expression
Balanced ExpressionStudent@project-lab:~/Udeet_200905406_C3/Week3$ |
```

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 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.

Filename: "q1_stack.h"

```
#define MAX 10
#define true 1
#define false 0

typedef struct
{
    char item[MAX];
    int top;
}stack;
```

```
void push(stack *ptr, char x);
void display(stack *ptr);
char pop(stack *ptr);
int isEmpty(stack *ptr);
int isFull(stack *ptr);
```

```
void push(stack *ptr, char x)
{
    if(!isFull(ptr))
    {
        ptr->top++;
        ptr->item[ptr->top]=x;
        printf("After push operation:\n");
        display(ptr);
    }
}
```

```
char pop(stack *ptr)
{
    if(!isEmpty(ptr))
    {
        printf("After pop operation:\n");
        char x=ptr->item[ptr->top--];
        display(ptr);
        return (x);
    }
}
```

```
int isEmpty(stack *ptr)
{
    if(ptr->top==-1)
        return true;
    else
        return false;
}
```

```
int isFull(stack *ptr)
{
    if(ptr->top==MAX-1)
        return 1;
    return 0;
}
```

```
void display(stack *ptr)
{
    for(int i=ptr->top; i>=0;i--)
    {
```

```

        printf("%c ",ptr->item[i]);
    }
    printf("\n");
}

```

Filename: “q1.c”

```

#include <stdio.h>
#include <stdlib.h>
#include "q1_stack.h"

int main()
{
    stack s;
    stack* st = &s;
    st->top = -1;
    int f = 1;
    printf("Name: Udeet Mittal\nBatch: C3\nRoll Number: 64\n");
    do
    {
        printf("Enter:\n1 to Push \n2 to Pop \n3 to Display Stack \n4 to Exit Program \n");
        int choice;
        scanf("%d", &choice);
        switch (choice)
        {
            case 1:
                {
                    char c;
                    printf("Enter the Character to Push:\n");
                    scanf(" %c", &c);
                    push(st, c);}
                break;
            case 2:{
                char c = pop(st);
                if (c != '\0')
                    printf("The Popped Character is: %c\n", c);
                }break;
            case 3:
                display(st);
                break;
            case 4:
                f = 0;
                break;
            default:
                printf("Invalid choice\n");
        }
    }while (f);
}

```

```
return 0;
```

```
}
```

```
Student@project-lab: ~/Udeet_200905406_C3/Week3
File Edit View Search Terminal Help
Student@project-lab:~/Udeet_200905406_C3/Week3$ gcc q1.c
Student@project-lab:~/Udeet_200905406_C3/Week3$ ./a.out
Name: Udeet Mittal
Batch: C3
Roll Number: 64
Enter:
1 to Push
2 to Pop
3 to Display Stack
4 to Exit Program
1
Enter the Character to Push:
A
After push operation:
A
Enter:
1 to Push
2 to Pop
3 to Display Stack
4 to Exit Program
1
Enter the Character to Push:
B
After push operation:
```

```
Student@project-lab: ~/Udeet_200905406_C3/Week3
File Edit View Search Terminal Help
2 to Pop
3 to Display Stack
4 to Exit Program
1
Enter the Character to Push:
B
After push operation:
B A
Enter:
1 to Push
2 to Pop
3 to Display Stack
4 to Exit Program
2
After pop operation:
A
The Popped Character is: B
Enter:
1 to Push
2 to Pop
3 to Display Stack
4 to Exit Program
4
Student@project-lab:~/Udeet_200905406_C3/Week3$
```

2. Convert a given decimal number to binary using stack

Filename: “q2_stack.h”

```
#define MAX 30
#define true 1
#define false 0
```

```
typedef struct
{
    int item[MAX];
    int top;
}stack;
```

```
void push(stack *ptr, int x);
```

```
void display(stack *ptr);
int pop(stack *ptr);
int isEmpty(stack *ptr);
int isFull(stack *ptr);
```

```

void push(stack *ptr,int x)
{
    if(!isFull(ptr))
    {
        ptr->top++;
        ptr->item[ptr->top]=x;
    }
}

int pop(stack *ptr)
{
    if(!isEmpty(ptr))
    {
        printf("After pop operation:\n");
        return ptr->item[ptr->top--];
    }
}

int isEmpty(stack *ptr)
{
    if(ptr->top==-1)
        return true;
    else
        return false;
}

int isFull(stack *ptr)
{
    if(ptr->top==MAX-1)
        return 1;
    return 0;
}

void display(stack *ptr)
{
    for(int i=ptr->top; i>=0;i--)
    {
        printf("%d ",ptr->item[i]);
    }
    printf("\n");
}

```

Filename: “q2.c”

```
#include <stdio.h>
```

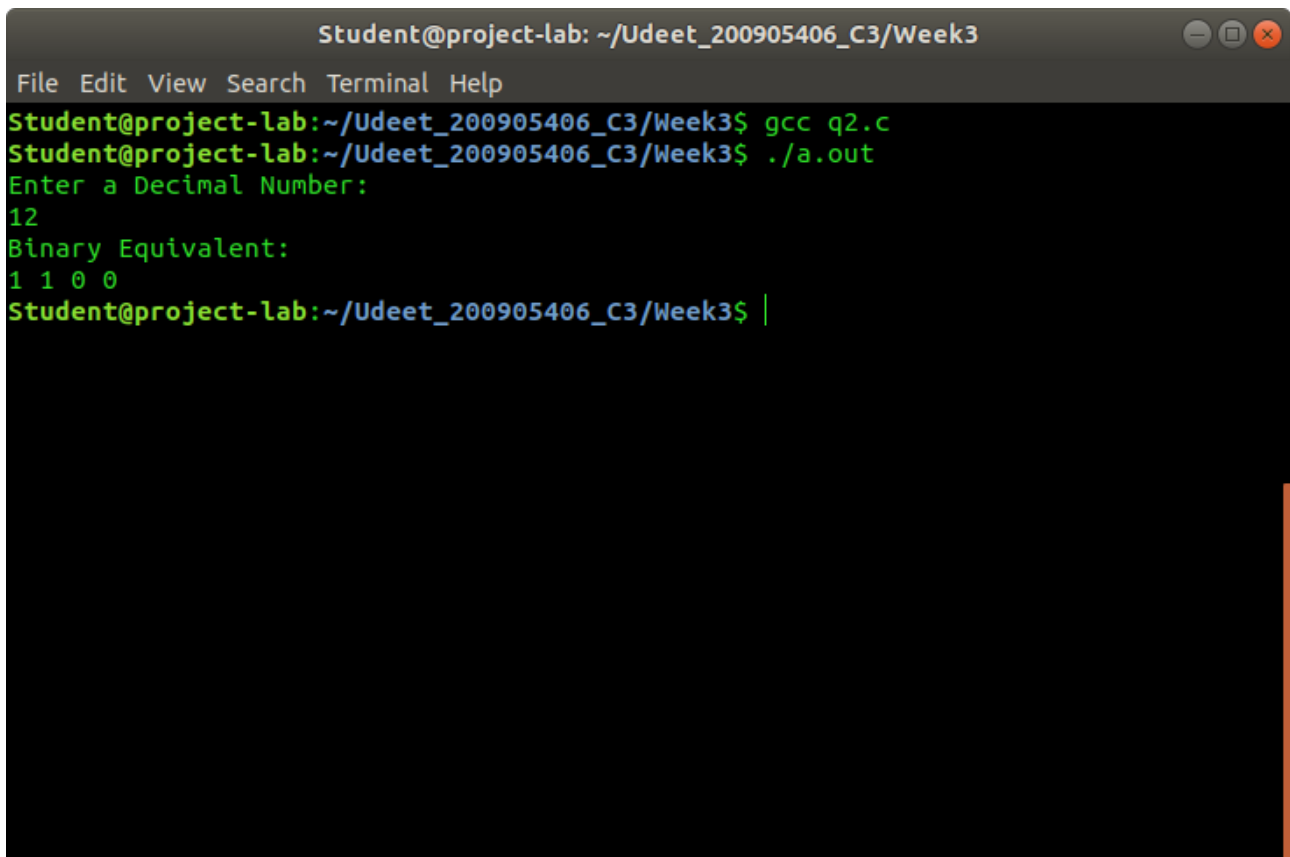


```

#include <stdlib.h>
#include "q2_stack.h"

int main()
{
    stack s;
    stack* st = &s;
    st->top = -1;
    printf("Name: Udeet Mittal\nBatch: C3\nRoll Number: 64\n");
    int n;
    printf("Enter a Decimal Number:\n");
    scanf("%d",&n);
    while(n!=0)
    {
        int d=n%2;
        push(st,d);
        n/=2;
    }
    printf("Binary Equivalent:\n");
    display(st);
    return 0;
}

```



```

Student@project-lab: ~/Udeet_200905406_C3/Week3
File Edit View Search Terminal Help
Student@project-lab:~/Udeet_200905406_C3/Week3$ gcc q2.c
Student@project-lab:~/Udeet_200905406_C3/Week3$ ./a.out
Enter a Decimal Number:
12
Binary Equivalent:
1 1 0 0
Student@project-lab:~/Udeet_200905406_C3/Week3$ |

```

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

Filename: “q3_stack.h”

```
#define MAX 100
#define true 1
#define false 0

typedef struct
{
    char item[MAX];
    int top;
}stack;

void push(stack *ptr, char x);
void display(stack *ptr);
char pop(stack *ptr);
int isEmpty(stack *ptr);
int isFull(stack *ptr);

void push(stack *ptr,char x)
{
    if(!isFull(ptr))
    {
        ptr->top++;
        ptr->item[ptr->top]=x;
    }
}

char pop(stack *ptr)
{
    if(!isEmpty(ptr))
        return ptr->item[ptr->top--];
}

int isEmpty(stack *ptr)
{
    if(ptr->top==-1)
        return true;
    else
        return false;
}

int isFull(stack *ptr)
{
    if(ptr->top==MAX-1)
        return 1;
    return 0;
}
```

```

}

void display(stack *ptr)
{
    for(int i=ptr->top; i>=0;i--)
    {
        printf("%c ",ptr->item[i]);
    }
    printf("\n");
}

```

Filename: “q3.c”

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "q3_stack.h"
int checkPalindrome(stack *);

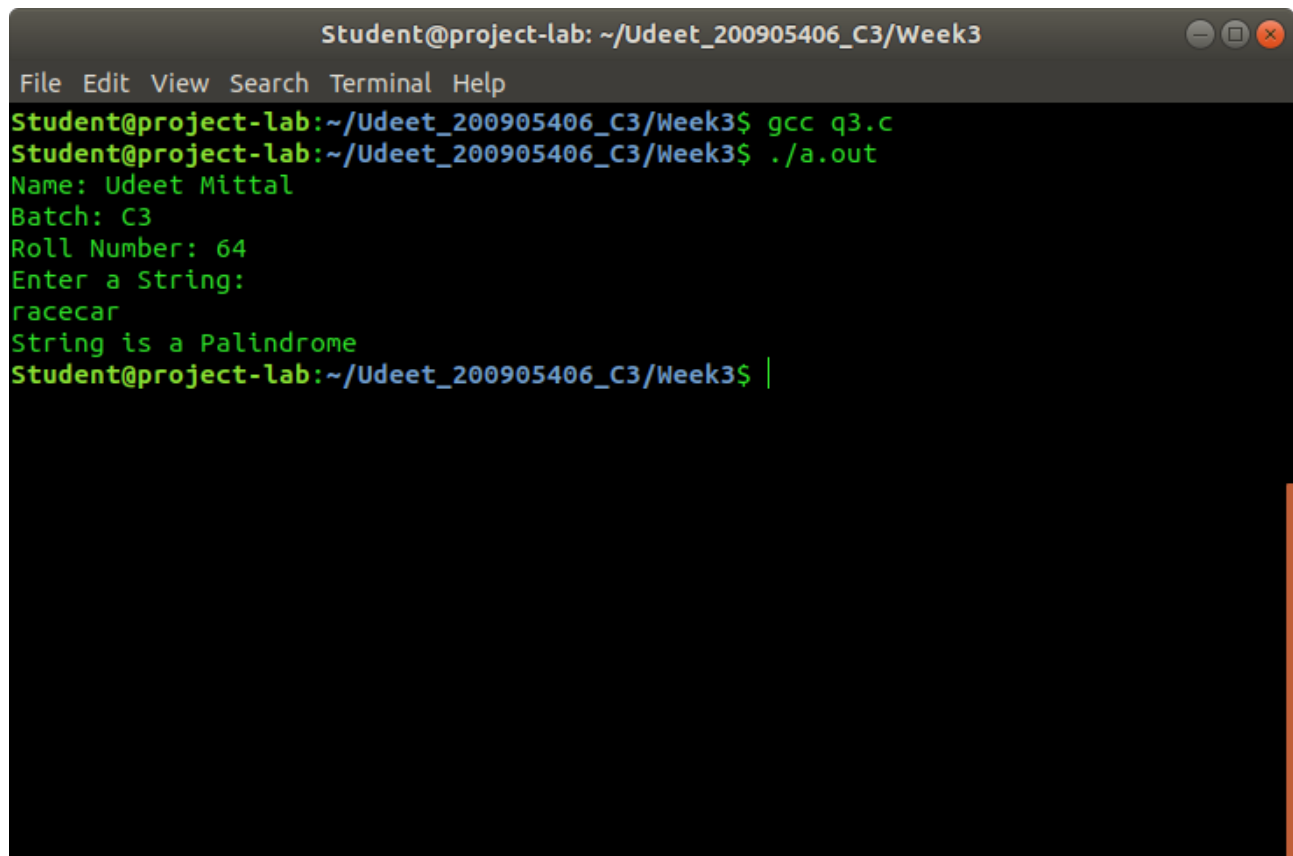
int main()
{
    printf("Name: Udeet Mittal\nBatch: C3\nRoll Number: 64\n");
    stack s;
    stack *st = &s;
    st->top = -1;
    printf("Enter a String:\n");
    scanf("%s", st->item);
    int check = checkPalindrome(st);
    if(check)
        printf("String is a Palindrome\n");
    else
        printf("String is Not a Palindrome\n");

    return 0;
}

int checkPalindrome(stack *s)
{
    int i;
    int length = strlen(s->item);
    for (i = 0; i < length / 2; i++)
    {
        push(s, s->item[i]);
    }
    if (length % 2)
        i++;
}

```

```
while (s->item[i] != '\0')
{
    char c = pop(s);
    if (c != s->item[i++])
        return 0;
}
return 1;
}
```

A terminal window titled "Student@project-lab: ~/Udeet_200905406_C3/Week3" with standard window controls. The terminal shows the execution of a C program. The user runs "gcc q3.c" and then "./a.out". The program prompts for "Name:", "Batch:", and "Roll Number:", which are answered as "Udeet Mittal", "C3", and "64" respectively. It then prompts "Enter a String:", where "racecar" is entered. The program outputs "String is a Palindrome" and returns to the prompt.

```
Student@project-lab: ~/Udeet_200905406_C3/Week3
File Edit View Search Terminal Help
Student@project-lab:~/Udeet_200905406_C3/Week3$ gcc q3.c
Student@project-lab:~/Udeet_200905406_C3/Week3$ ./a.out
Name: Udeet Mittal
Batch: C3
Roll Number: 64
Enter a String:
racecar
String is a Palindrome
Student@project-lab:~/Udeet_200905406_C3/Week3$ |
```