

DS LAB

Lab 5

Q1)

Source Code:

“charstack.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);
```

```
void display(stack *);
```

```
char pop(stack *ps);
```

```
int isEmpty(stack *ps);
```

```
int isFull(stack *ps);
```

```
void push(stack *ps, char x) {
```

```
    if (!isFull(ps))
```

```
    {
```

```
        ps->top++;
```

```
        ps->item[ps->top] = x;
```

```
    }
```

```
    printf("After push: \n");
```

```
    display(ps);
```

```
}
```

```
char pop(stack *ps)
```

```
{
```

```
    if (!isEmpty(ps)){
```

```
        ps->top--;
```

```
        printf("After pop: \n");
```

```
        display(ps);
```

```
        return(ps->item[ps->top]);
```

```
    }
```

```
}
```

```
int isEmpty(stack *ps)
```

```
{
```

```
    if (ps->top== -1)
```

```
        return(true);
```

```
    else
```

```
        return(false);
```

```
}
```

```

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

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

```

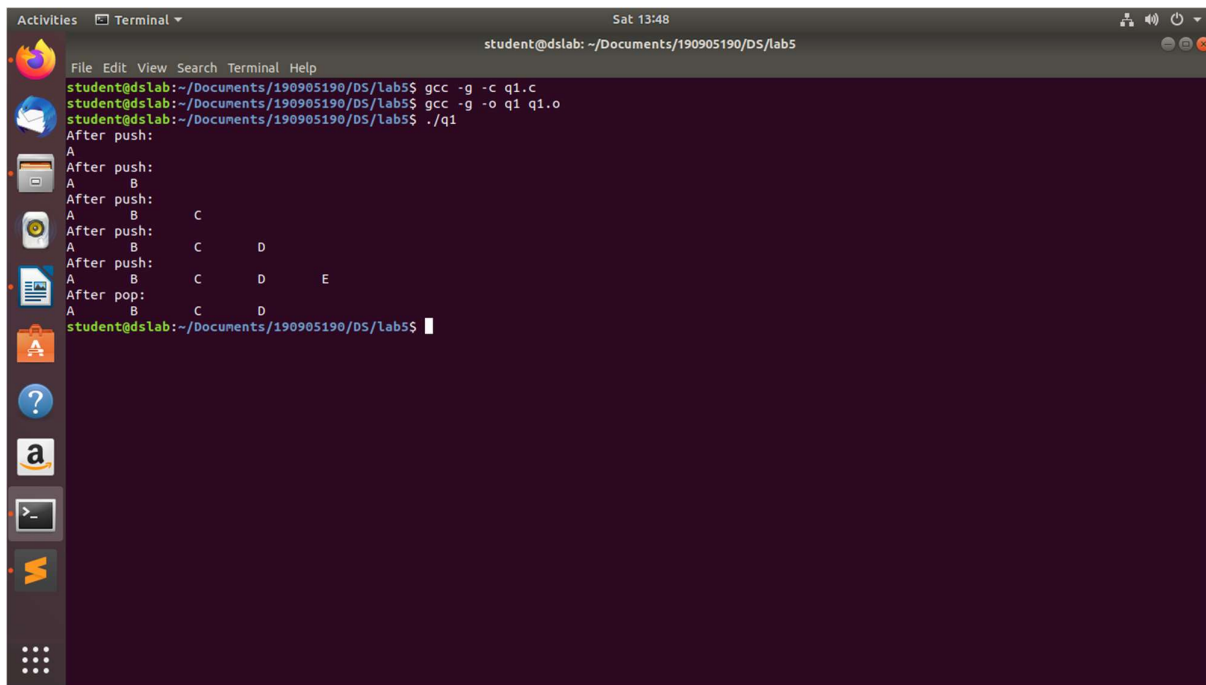
“q1.c”

```

#include <stdio.h>
#include <stdlib.h>
#include "charstack.h"
int main()
{
    stack *ps;
    ps = (stack *)malloc(sizeof(stack));
    ps->top = -1;
    // printf("Hello\n");
    push(ps, 'A');
    push(ps, 'B');
    push(ps, 'C');
    push(ps, 'D');
    push(ps, 'E');
    char ch = pop(ps);
    return 0;
}

```

Output:



```
student@ds1ab: ~/Documents/190905190/DS/lab5$ gcc -g -c q1.c
student@ds1ab:~/Documents/190905190/DS/lab5$ gcc -g -o q1 q1.o
student@ds1ab:~/Documents/190905190/DS/lab5$ ./q1
After push:
A
After push:
A B
After push:
A B C
After push:
A B C D
After push:
A B C D E
After pop:
A B C D
student@ds1ab:~/Documents/190905190/DS/lab5$
```

Q2)

Source Code:

“binstack.h”

```
# define MAX 30
```

```
# define true 1
```

```
# define false 0
```

```
/* Structure definition */
```

```
typedef struct
```

```
{
```

```
    int item[MAX];
```

```
    int top;
```

```
}stack;
```

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

```
void display(stack *);
```

```
int pop(stack *ps);
```

```
int isEmpty(stack *ps);
```

```
int isFull(stack *ps);
```

```
void push(stack *ps, int x) {
```

```
    if (!isFull(ps))
```

```
    {
```

```
        ps->top++;
```

```
        ps->item[ps->top] = x;
```

```
    }
```

```
}
```

```
int pop(stack *ps)
```

```
{
```

```
    if (!isEmpty(ps)){
```

```
        ps->top--;
```

```
        return(ps->item[ps->top]);
```

```

    }
}

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

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

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

```

“q2.c”

```

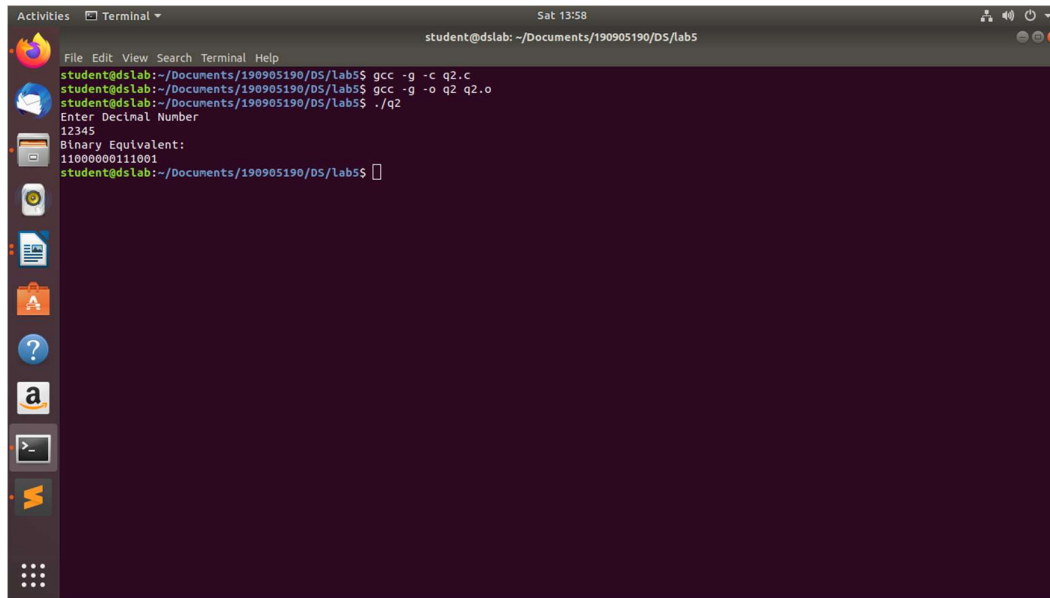
#include <stdio.h>
#include <stdlib.h>
#include "binstack.h"

int main()
{
    int N;
    printf("Enter Decimal Number\n");
    scanf("%d", &N);
    stack * bin;
    bin = (stack *)malloc(sizeof(stack));
    bin->top = -1;

    while(N!=0)
    {
        int d = N%2;
        push(bin, d);
        N = N/2;
    }
    printf("Binary Equivalent: \n");
    display(bin);
    return 0;
}

```

Output:



A terminal window titled "Terminal" with a menu bar (File, Edit, View, Search, Terminal, Help) and a status bar (Sat 13:58, student@dslab: ~/Documents/190905190/DS/lab5). The terminal shows the following commands and output:

```
student@dslab:~/Documents/190905190/DS/lab5$ gcc -g -c q2.c
student@dslab:~/Documents/190905190/DS/lab5$ gcc -g -o q2 q2.o
student@dslab:~/Documents/190905190/DS/lab5$ ./q2
Enter Decimal Number
12345
Binary Equivalent:
11000000111001
student@dslab:~/Documents/190905190/DS/lab5$
```

The terminal window has a dark purple background. On the left side, there is a vertical dock with several application icons: a red and yellow icon, a blue and white icon, a yellow and black icon, a blue and white icon, an orange icon, a blue question mark icon, an Amazon logo, a terminal icon, and a yellow icon. The status bar at the top right shows system icons for network, sound, and power.

Q3)

Source Code:

“palinstack.h”

```
# define MAX 100
# define true 1
# define false 0
/* Structure definition */
typedef struct
{
    char item[MAX];
    int top;
} stack;

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

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

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

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

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

void display(stack *ps)
```

```

{
    for (int i = 0; i <= ps->top; ++i)
    {
        printf("%c\t", ps->item[i]);
    }
    printf("\n");
}

int checkPalin(stack *ps)
{
    for (int i = 0; i <= ps->top; ++i)
    {
        if(ps->item[i] != ps->item[ps->top - i])
            return 0;
    }
    return 1;
}

```

“q3.c”

```

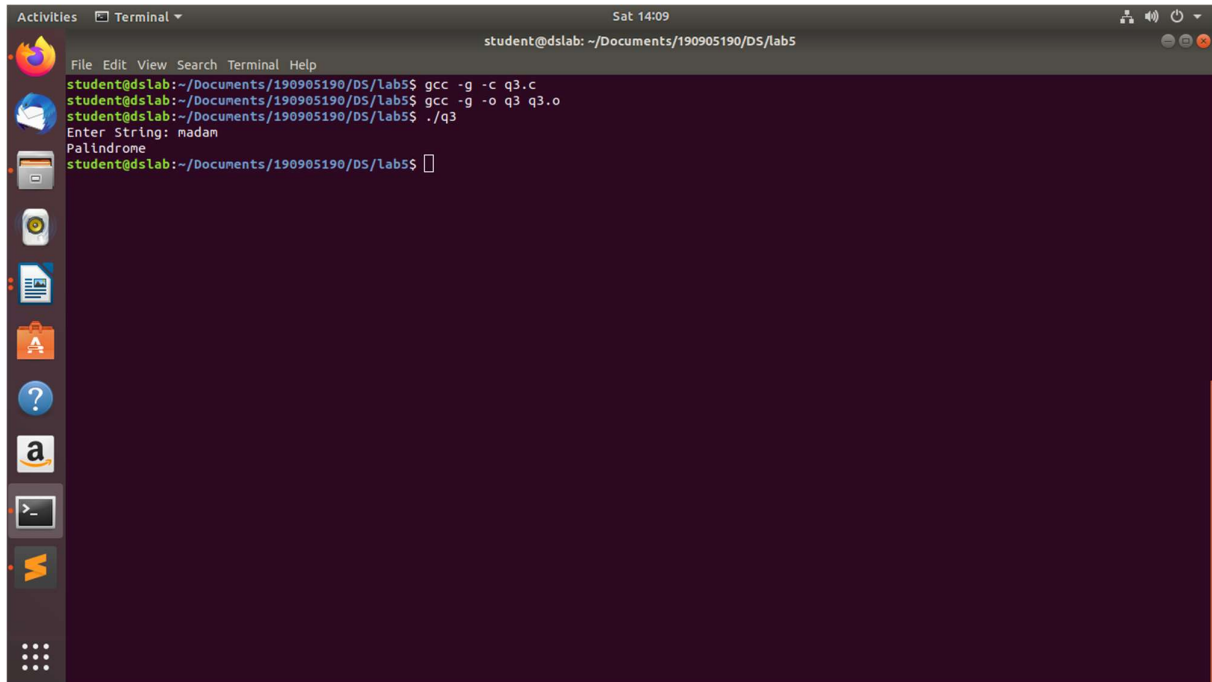
#include <stdio.h>
#include <stdlib.h>
#include "palinstack.h"
#include <string.h>

int main()
{
    char str[MAX];
    stack * strs;
    strs = (stack *)malloc(sizeof(stack));
    strs->top = -1;

    printf("Enter String: ");
    scanf("%s", str);
    for(int i = 0; i < strlen(str); i++)
    {
        push(strs, str[i]);
    }
    int f = checkPalin(strs);
    if(f == 0)
        printf("Not a Palindrome\n");
    else
        printf("Palindrome\n");
    return 0;
}

```

Output:



A terminal window titled "Terminal" with a menu bar (File, Edit, View, Search, Terminal, Help) and a status bar (Sat 14:09, student@dslab: ~/Documents/190905190/DS/lab5). The terminal shows the following commands and output:

```
student@dslab:~/Documents/190905190/DS/lab5$ gcc -g -c q3.c
student@dslab:~/Documents/190905190/DS/lab5$ gcc -g -o q3 q3.o
student@dslab:~/Documents/190905190/DS/lab5$ ./q3
Enter String: madam
Palindrome
student@dslab:~/Documents/190905190/DS/lab5$
```

The terminal window has a dark purple background and a vertical sidebar on the left with various application icons. The output of the program is "madam" followed by "Palindrome".