Assignment – 05

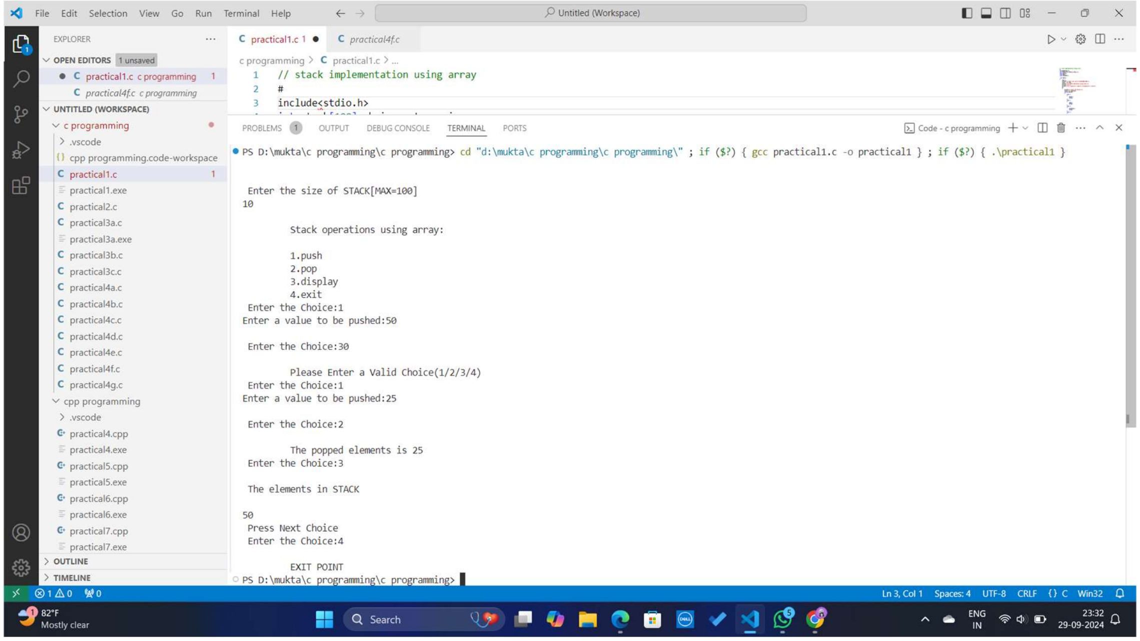
Q. Implement a Stack and perform the stack operations: Push, Pop and Print using Menu Driver Program such as 1.Push, 2.Pop and 3. Print and 4. Exit using array .

Program :

```
// stack implementation using array

#

include<stdio.h>

int stack[100],choice,n,top,x,i;

void push();

void pop();

void display();


 int main()

{

    top=-1;

    printf("\n Enter the size of STACK[MAX=100]");

    scanf ("%d",&n);

    printf("\n\t Stack operations using array:");

    printf("\n\t");

    printf("\n\t 1.push\n\t 2.pop\n\t 3.display\n\t 4.exit");

    do

    {

        printf("\n Enter the Choice:");

        scanf("%d",&choice);

        switch(choice)

        {

            case 1:

            {

                push();

                break;

            }
            case 2:
            {
                pop();
                break;
            }
            case 3:
            {
                display();
                break;
            }
            case 4:
            {
                printf("\n\t EXIT POINT ");
                break;
            }
            default:
            {
                printf ("\n\t Please Enter a Valid Choice(1/2/3/4)");
            }

        }
    }
    while(choice!=4);
    return 0;
}
void push()
{
    if(top>=n-1)
    {
        printf("\n\tSTACK is over flow");
    }
    else
    {
        printf("Enter a value to be pushed:");
        scanf("%d",&x);
        top++;
        stack[top]=x;
    }
}
void pop()
{
    if(top<=-1)
    {
        printf("\n\t Stack is under flow");
    }
    else
    {
        printf("\n\t The popped elements is %d",stack[top]);
        top--;
    }
}
void display()
{
    if(top>=0)
    {
        printf("\n The elements in STACK \n");
        for(i=top; i>=0; i--)
            printf("\n%d",stack[i]);
        printf("\n Press Next Choice");
    }
    else
```

```
   {
      printf("\n The STACK is empty");
   }
}
```

OUTPUT:



```
// stack implementation using array
#
include<stdio.h>
```

```
PS D:\mukta\c programming\c programming> cd "d:\mukta\c programming\c programming\" ; if ($?) { gcc practical1.c -o practical1 } ; if ($?) { .\practical1 }


 Enter the size of STACK[MAX=100]
10

         Stack operations using array:

         1.push
         2.pop
         3.display
         4.exit
 Enter the Choice:1
Enter a value to be pushed:50

 Enter the Choice:30

         Please Enter a Valid Choice(1/2/3/4)
 Enter the Choice:1
Enter a value to be pushed:25

 Enter the Choice:2

         The popped elements is 25
 Enter the Choice:3

 The elements in STACK

50
 Press Next Choice
 Enter the Choice:4

         EXIT POINT
PS D:\mukta\c programming\c programming>
```

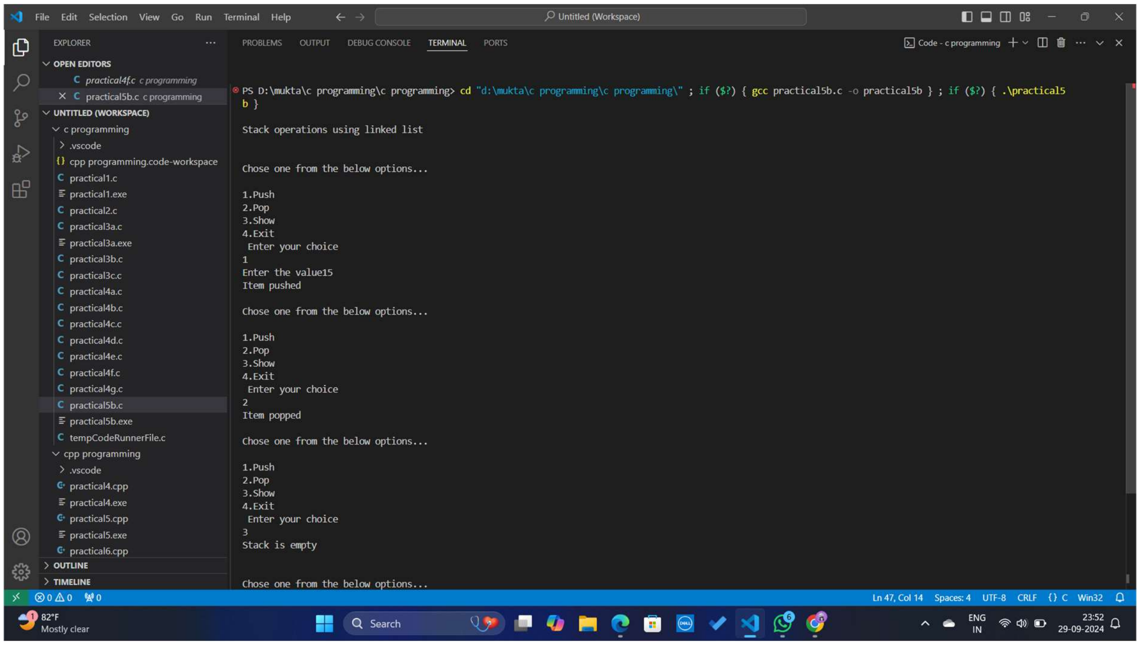
Q. Implement stack using linked list.

Program :

```
//stack implementation using linked list
#include <stdio.h>
#include <stdlib.h>
void push();
void pop();
void display();
struct node
{
int val;
struct node *next;
```

```
};
struct node *head;

void main ()
{
    int choice=0;
    printf("\nStack operations using linked list\n");

    while(choice != 4)
    {
        printf("\n\nChose one from the below options...\n");
        printf("\n1.Push\n2.Pop\n3.Show\n4.Exit");
        printf("\n Enter your choice \n");
        scanf("%d",&choice);
        switch(choice)
        {
            case 1:
            {
                push();
                break;
            }
            case 2:
            {
                pop();
                break;
            }
            case 3:
            {
                display();
                break;
            }
            case 4:
            {
                printf("Exit code");
                break;
            }
            default:
            {
                printf("Please Enter valid choice ");
            }
    };
}
}
void push ()
{
    int val;
    struct node *ptr = (struct node*)malloc(sizeof(struct node));
    if(ptr == NULL)
    {
        printf("not able to push the element");
    }
    else
    {
        printf("Enter the value");
        scanf("%d",&val);
        if(head==NULL)
        {
            ptr->val = val;
            ptr -> next = NULL;
            head=ptr;
        }
        else
        {
            ptr->val = val;
            ptr->next = head;
            head=ptr;

        }
        printf("Item pushed");

    }
}

void pop()
{
    int item;
    struct node *ptr;
    if (head == NULL)
    {
        printf("Underflow");
    }
    else
    {
        item = head->val;
        ptr = head;
        head = head->next;
        free(ptr);
        printf("Item popped");

    }
}
void display()
{
    int i;
    struct node *ptr;
    ptr=head;
    if(ptr == NULL)
    {
        printf("Stack is empty\n");
    }
    else
    {
        printf("Printing Stack elements \n");
        while(ptr!=NULL)
        {
            printf("%d\n",ptr->val);
            ptr = ptr->next;
        }
    }
}
```

OUTPUT:



```
PS D:\mukta\c programming\c programming> cd "d:\mukta\c programming\c programming\" ; if ($?) { gcc practical5b.c -o practical5b } ; if ($?) { .\practical5
b }

Stack operations using linked list


Chose one from the below options...

1.Push
2.Pop
3.Show
4.Exit
 Enter your choice
1
Enter the value15
Item pushed

Chose one from the below options...

1.Push
2.Pop
3.Show
4.Exit
 Enter your choice
2
Item popped

Chose one from the below options...

1.Push
2.Pop
3.Show
4.Exit
 Enter your choice
3
Stack is empty


Chose one from the below options...
```