NAME : IBRAHIM HAJI FAKI

REG NO : BCS/18/23/028/TZ

ASSIGNMENT : Cychelesheet 1

Github username : ibrahimhajifaki

Git repository link : https://github.com/ibrahimhajifaki/ibrahimhajifaki.git

QUESTION 1 :

Write a Program to implement stack with the following operations

i. Push
ii. Pop
iii. Display

ANSWER :

```
#include <stdio.h>

#define size 15

int stack[size];

int top=-1;


void push(int x)

{

    if(top==size-1)

    {

            printf("The stackis full ");

    }

    else

    {

            top++;

            stack[top]=x;

            printf("The value %d is pushed into the stack " ,x);
    }
    printf("\n");
}

int pop()
{
    if(top==-1)
    {
            printf("The stack is empty ");
            return -1;
    }
    else
    {
            int x;
            x=stack[top];
            top--;
            return x;
            printf("The deleted value is %d" ,x);
    }

}

int peek()
{
    if(top==-1)
    {
            printf("The stack is empty ");
            return -1;
    }
    else
    {
            return stack[top];
    }
}

void display()
{
    if(top==-1)
    {
            printf("The stack is empty");
    }
    else
    {
            for(int i=top;i>=0;i--)
            {
                    printf("%d " ,stack[i]);
            }
            printf("\n");
    }
}

int main()
{
    int choice=1;
    int value;
    while(choice==1)
    {
            printf("1 . Push \n");
            printf("2 . Pop \n");
            printf("3 . Peek \n");
            printf("4 . Dispaly \n");
            printf("Enter yuour choice ");
```

```
        scanf("%d" ,&choice);
        switch(choice)
        {
                case 1:
                        printf("Enter number to add :");
                        scanf("%d",&value);
                        push(value);
                        break;
                case 2:
                        value=pop();
                        if(value!=-1)
                        {
                                printf("The deleted value is %d" ,value);
                                printf("\n");
                        }
                        break;
                case 3:
                        value=peek();
                        if(value!=-1)
                        {
                                printf("the top value is %d" ,value);
                                printf("\n");
                        }
                        break;
                case 4:
                        display();
                        break;
        }
        printf("do you want to continue ? (1/0)");
        scanf("%d" ,&choice);
}
```

```
}
```

OUTPUT

QUESTION 2:

Write a C program to implement to infix to postifix operation

ANSWER :

```
#include <stdio.h>
#define size 15
int stack[size];
int top=-1;

void push(int x)
{
    if(top==size-1)
    {
            printf("The stackis full ");
    }
    else
    {
            top++;
            stack[top]=x;
            printf("The value %d is pushed into the stack " ,x);
    }
    printf("\n");
}

int pop()
{
    if(top==-1)
    {
            printf("The stack is empty ");
            return -1;
    }
    else
    {
        int x;
        x=stack[top];
        top--;
        return x;
        printf("The deleted value is %d" ,x);
    }

}

int peek()
{
    if(top==-1)
    {
        printf("The stack is empty ");
        return -1;
    }
    else
    {
        return stack[top];
    }
}

void display()
{
    if(top==-1)
    {
        printf("The stack is empty");
    }
    else
    {
            for(int i=top;i>=0;i--)
            {
                    printf("%d " ,stack[i]);
            }
            printf("\n");
    }
}

int main()
{
    int choice=1;
    int value;
    while(choice==1)
    {
            printf("1 . Push \n");
            printf("2 . Pop \n");
            printf("3 . Peek \n");
            printf("4 . Dispaly \n");
            printf("Enter yuour choice ");
            scanf("%d" ,&choice);
            switch(choice)
            {
                    case 1:
                            printf("Enter number to add :");
                            scanf("%d",&value);
                            push(value);
                            break;
                    case 2:
                            value=pop();
                            if(value!=-1)
```

```
                        {
                                printf("The deleted value is %d" ,value);
                                printf("\n");
                        }
                        break;
                case 3:
                        value=peek();
                        if(value!=-1)
                        {
                                printf("the top value is %d" ,value);
                                printf("\n");
                        }
                        break;
                case 4:
                        display();
                        break;
        }
        printf("do you want to continue ? (1/0)");
        scanf("%d" ,&choice);
    }
}
```

OUT PUT

QUESTION 3:

Write a C program to implement infix to prefix expression

ANSWER :

```
#include <stdio.h>

#include <ctype.h>

#include <string.h>

#define MAX 100


char stack[MAX];

int top = -1;


int precedence(char c)

{

    if (c == '+' || c == '-')

    {

        return 1;

    }

    if (c == '*' || c == '/')

    {

        return 2;

    }

    return 0;

}


void reverse(char exp[])

{

    int length = strlen(exp);

    for (int i = 0; i < length / 2; i++)

    {

        char temp = exp[i];
        exp[i] = exp[length - i - 1];
        exp[length - i - 1] = temp;
    }
}

void infixToPrefix(char infix[], char prefix[])
{
    reverse(infix);
    int i = 0, k = 0;
    while (infix[i] != '\0')
    {
        if (isalnum(infix[i]))
        {
            prefix[k++] = infix[i];
        }
        else if (infix[i] == ')')
        {
            stack[++top] = infix[i];
        }
        else if (infix[i] == '(')
        {
            while (top != -1 && stack[top] != ')')
            {
                prefix[k++] = stack[top--];
            }
            top--;
        }
        else
        {
            while (top != -1 && precedence(stack[top]) >= precedence(infix[i]))
            {
                prefix[k++] = stack[top--];
            }
            stack[++top] = infix[i];
        }
        i++;
    }

    while (top != -1)
    {
        prefix[k++] = stack[top--];
    }

    prefix[k] = '\0';
    reverse(prefix);
}

int main()
{
    char infix[MAX], prefix[MAX];
    printf("Enter infix expression: ");
    scanf("%s", infix);
    infixToPrefix(infix, prefix);
    printf("Prefix expression: %s\n", prefix);
    return 0;
}
```

OUTPUT

QUESTION 4:

Write a C program to implement evaluation of prefix

ANSWER :

```
#include <stdio.h>

#include <ctype.h>

#include <string.h>

#include <stdlib.h>

#define MAX 100


int stack[MAX];
int top = -1;

int evaluatePrefix(char prefix[])
{
    int i = strlen(prefix) - 1;

    while (i >= 0)
    {
        if (isdigit(prefix[i]))
        {
            stack[++top] = prefix[i] - '0';
        }
        else
        {
            int operand1 = stack[top--];
            int operand2 = stack[top--];

            switch (prefix[i])
            {
                case '+': stack[++top] = operand1 + operand2; break;
                case '-': stack[++top] = operand1 - operand2; break;
                case '*': stack[++top] = operand1 * operand2; break;
                case '/': stack[++top] = operand1 / operand2; break;
                case '^': stack[++top] = operand1 ^ operand2; break;
            }
        }
        i--;
    }

    return stack[top];
}
```

```
int main()
{
    char prefix[MAX];

    printf("Enter prefix expression: ");
    scanf("%s", prefix);

    int result = evaluatePrefix(prefix);
    printf("Result of evaluation: %d\n", result);

    return 0;
}
```

OUTPUT :

QUESTION 5 :

Write a C program to implement evaluation of postifix

ANSWER :

```
#include <stdio.h>
#include <ctype.h>
#include <string.h>
#include <stdlib.h>
#define MAX 100

int stack[MAX];
int top = -1;

int evaluatePostfix(char postfix[])
{
    int i = 0;

    while (postfix[i] != '\0')
    {
        if (isdigit(postfix[i]))
        {
            stack[++top] = postfix[i] - '0';
        }
        else
        {
            int operand2 = stack[top--];
            int operand1 = stack[top--];

            switch (postfix[i])
            {
                case '+': stack[++top] = operand1 + operand2; break;
                case '-': stack[++top] = operand1 - operand2; break;
                case '*': stack[++top] = operand1 * operand2; break;
                case '/': stack[++top] = operand1 / operand2; break;
                case '^': stack[++top] = operand1 ^ operand2; break;
            }
        }
        i++;
    }

    return stack[top];
}

int main()
{
    char postfix[MAX];

    printf("Enter postfix expression: ");
    scanf("%s", postfix);

    int result = evaluatePostfix(postfix);
    printf("Result of evaluation: %d\n", result);

    return 0;
}
```

OUTPUT :

QUESTION 6:

Write a C program to implement linear queue using array with following operations

i. Insertion
ii. Deletion
iii. Display

ANSWER :

```
#include <stdio.h>

#define size 10


struct queue

{

      int que[size];

      int front;

      int rear;

}q;


void init()

{

      q.front=q.rear=-1;

}


int full()

{
    if(q.rear>=size-1)
            return 1;
    else
            return 0;
}

int empty()
{
    if(q.rear==-1 || q.front>q.rear)
            return 1;
    else
            return 0;
}

int insert(int item)
{
    if(q.front==-1)
    {
            q.front++;
            q.que[q.rear]=item;
            return q.rear;
    }
}

int delete()
{
    int item;
    item=q.que[q.front];
    q.front++;
    printf("The deleted element is %d" ,item);
    return q.front;
```

```
}

void display()
{
    for(int i=q.front;i<q.rear;i++)
    {
            printf("%d ",q.que[i]);
    }
}

int main()
{
    int choice=1;
    int element;
    while(choice==1)
    {
            printf(" 1. Insert \n");
            printf(" 2. Delete \n");
            printf(" 3. Display \n");
            printf("Enter your choice :");
            scanf("%d",&choice);
            switch(choice)
            {
                    case 1:
                            printf("Enter elements of que ");
                            scanf("%d",&element);
                            insert(element);
                            break;
                    case 2:
                            element = delete();
                            printf("The element %d is deleted from the queue");
```

```
                        break;
                case3:
                        display();
                        break;
            }
            printf("Do you want to continue ? (1/0)");
            scanf("%d",&choice);
    }
}
```

OUTPUT :