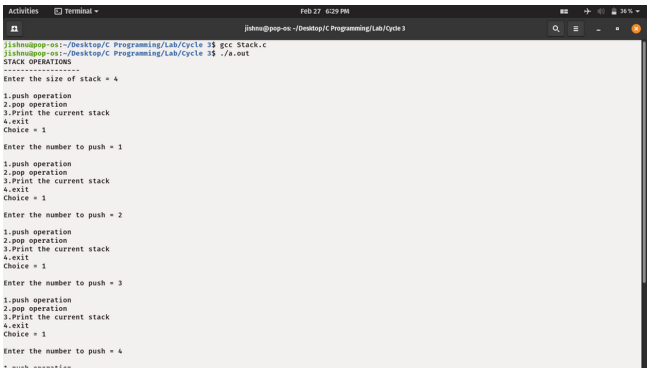
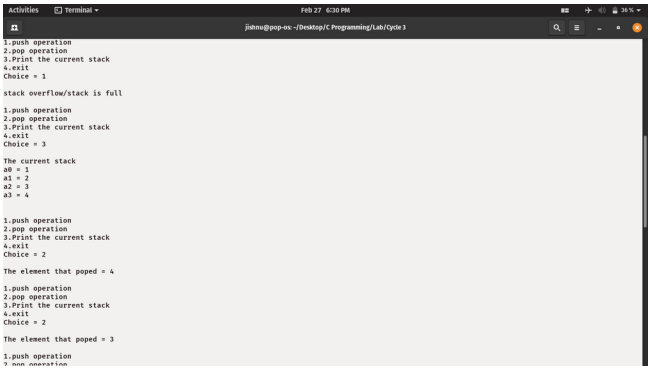
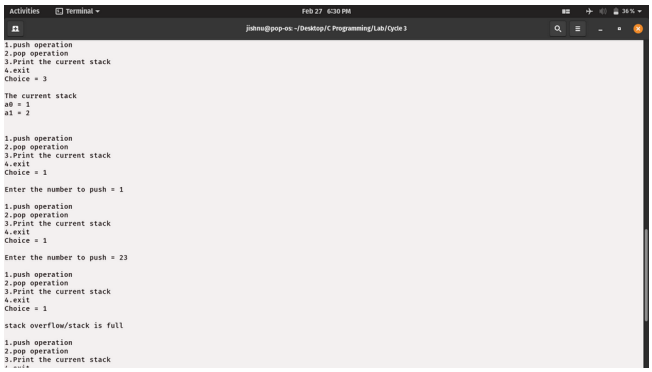
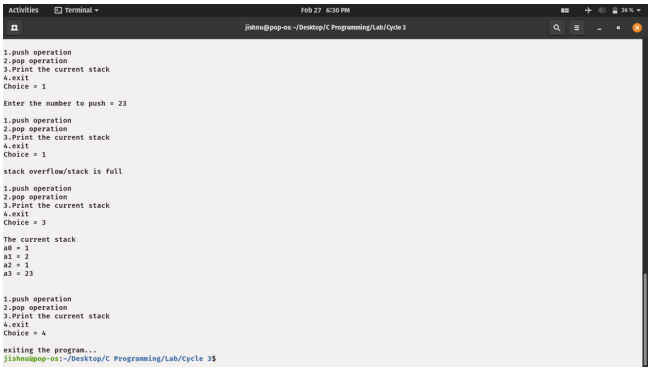
(1) STACK

```
#include<stdio.h>
#include<stdlib.h>
void main(){
	int stack[10];
	int top=-1;
	int item;
	int i,max;
	int op=1;
	printf("STACK OPERATIONS\n");
	for(i=0;i<18;i++){
		printf("%c",'-');
	}printf("\n");
	printf("Enter the size of stack = ");
	scanf("%d",&max);
	while(op<4){
		printf("\n1.push operation\n2.pop operation\n3.Print the current stack\n4.exit\n");
		printf("Choice = ");
		scanf("%d",&op);
		printf("\n");
		switch(op){
		case 1 : if(top == max-1){
				printf("stack overflow/stack is full\n");
				break;
			}else{
				printf("Enter the number to push = ");
				scanf("%d",&item);
				top=top+1;
				stack[top]=item;
			}break;
		case 2 : if(top == -1){
				printf("stack underflow/stack is empty\n");
				break;
			}else{
				item=stack[top];
				top=top-1;
				printf("The element that poped = %d\n",item);
				break;
			}break;
		case 3 : printf("The current stack\n");
			if(top == -1){
				printf("stack underflow/empty\n");
				break;
			}else{
				for(i=0;i<=top;i++){
					printf("a%d = %d\n",i,stack[i]);
				}
				printf("\n");
			}break;
		case 4 : printf("exiting the program...\n");
			exit(0);
		default:printf("Something wrong!!!\nexiting the program\n");
			exit(0);
	}}
}
```









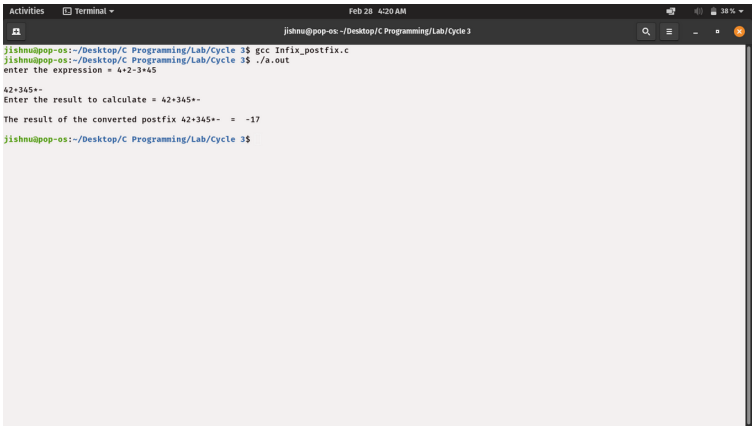
(2) INFIX POSTFIX

```
#include <stdio.h>
#include <ctype.h>
char stack[100];
int top = -1;

void push(char x) {
stack[++top] = x;
}
char pop() {
if(top == -1)
    return -1;
else
    return stack[top--]; } char pop2(){
return stack[top--]; } int priority(char x) {
```

```
if(x == '(')
   return 0;
if(x == '+' || x == '-')
   return 1;
if(x == '*' || x == '/')
   return 2;
return 0;
 }
void main() {
char exp[20];
char *e, x;
printf("enter the expression : ");
scanf("%s", exp);
printf("\n");
e = exp;
while (*e != '\0')
{
   if(isalnum(*e))
      printf("%c", *e);
   else if (*e == '(')
      push(*e);
   else if(*e == ')')
   {
      while ((x = pop()) != '(')
         printf("%c", x);
   }
   else{
      while(priority(stack[top]) >= priority(*e))
         printf("%c", pop());
      push(*e);
   }
   e++;
   }
while(top != -1)
{
   printf("%c", pop());
}
char Postfix[20];
char *P;
int n1,n2,n3,num;
 printf("\nEnter the result to calculate :: ");
 scanf("%s",Postfix);
   P = Postfix;
   while(*P != '\0')
   {
      if(isdigit(*P))
      {
         num = *P - 48;
         push(num);
      }
      else
      {
         n1 = pop2();
         n2 = pop2();
         switch(*P)
         {
         case '+':
         {
            n3 = n1 + n2;
            break;
```

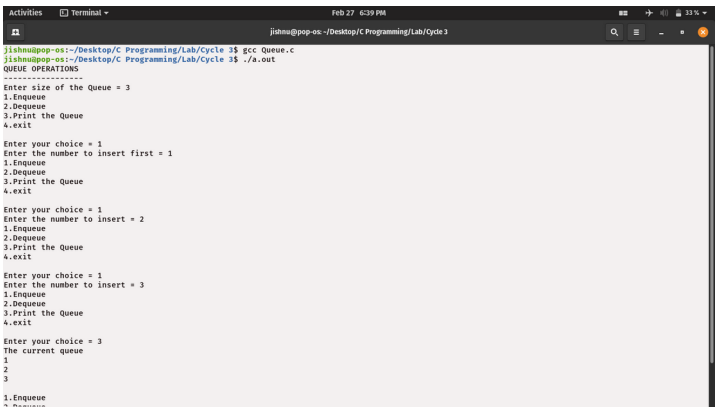
```
            }
            case '-':
            {
               n3 = n2 - n1;
               break;
            }
            case '*':
            {
               n3 = n1 * n2;
               break;
            }
            case '/':
            {
               n3 = n2 / n1;
               break;
            }
            }
            push(n3);
         }
         P++;
      }
      printf("\nThe result of the converted postfix %s  =  %d\n\n",Postfix,pop2());
}
```

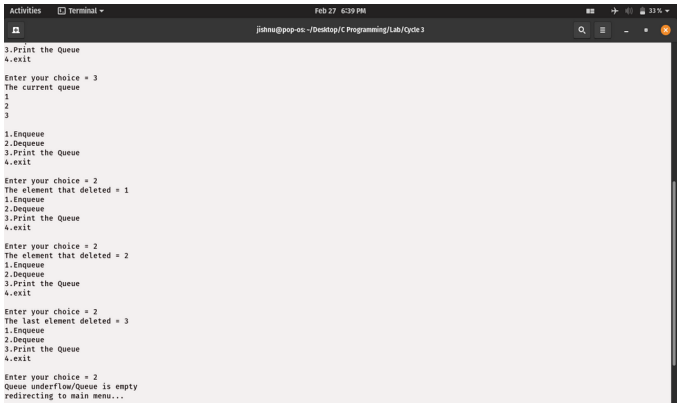


(3) QUEUE

```
#include<stdio.h>
#include<stdlib.h>
void main(){
        int i,max;
        int Q[100];
        int front = -1;
        int rear = -1;
        int item;
        int ch=1;
        printf("QUEUE OPERATIONS\n");
        for(i=0;i<17;i++){
                printf("%c",'-');
        }
        printf("\n");
        printf("Enter size of the Queue = ");
        scanf("%d",&max);
        while(ch<5){
                printf("1.Enqueue\n2.Dequeue\n3.Print the Queue\n4.exit\n\n");
                printf("Enter your choice = ");
                scanf("%d",&ch);
```

```
		switch(ch){
			case 1 : if(rear == max-1){
					printf("Queue overflow/Queue ended\nredirecting to main menu...\n\
n");
					break;
				}
				else if(front==-1 && rear==-1){
					front=rear=0;
					printf("Enter the number to insert first = ");
					scanf("%d",&item);
					Q[rear]=item;
				}
				else{
					rear = rear+1;
					printf("Enter the number to insert = ");
					scanf("%d",&item);
					Q[rear] = item;
				}
				break;
			case 2 : if(front==-1 && rear==-1){
					printf("Queue underflow/Queue is empty\nredirecting to main menu...\
n\n");
				}
				else if(front==rear){
					item = Q[front];
					printf("The last element deleted = %d\n",Q[front]);
					front=rear=-1;
				}
				else{
					item = Q[front];
					printf("The element that deleted = %d\n",Q[front]);
					front = front+1;
				}
				break;
			case 3 : if(front == -1){
					printf("Queue is empty\nredirecting to main menu...\n\n");
					break;
				}
				else{
					printf("The current queue\n");
					for(i=front;i<=rear;i++){
						printf("%d\n",Q[i]);
					}
				}
				printf("\n");
				break;
			case 4 : printf("exiting the program...\n");
				exit(0);
			default: printf("Something went wrong !!!\nprogram terminated...\n");
				exit(0);
		}
	}
}
```





(4) CIRCULAR QUEUE

```
#include<stdio.h>
#include<stdlib.h>
void main(){
        int i,max;
        int op=1;
        int front=-1;
        int rear=-1;
        int item;
        int Q[max];
        printf("CIRCULAR QUEUE OPERATIONS\n");
        for(int j=0;j<25;j++){
                printf("%c",'-');
        }
        printf("\n");
        printf("Enter size of circular Queue = ");
        scanf("%d",&max);
        while(op<4){
                printf("1.Enqueue\n2.Dequeue\n3.Print CircularQ\n4.exit\n\n");
                printf("Choice = ");
                scanf("%d",&op);
                switch(op){
                        case 1 : if((rear+1)%max == front){
                                        printf("Queue overflow/full\n");
                                        break;
                                }else if(front==-1){
                                        front=rear=0;
                                        printf("Enter the number to insert = ");
                                        scanf("%d",&item);
                                        Q[rear]=item;
                                }else{
                                        rear = (rear+1)%max;
                                        printf("Enter the element to insert = ");
                                        scanf("%d",&item);
                                        Q[rear]=item;
                                }break;
                        case 2 : if(front==-1){
                                        printf("Queue underflow/empty\n");
                                }else if(front==rear){
```

```
                        item=Q[front];
                        printf("The last element deleted = %d\n",Q[front]);
                        front=rear=-1;
                    }else{
                        item=Q[front];
                        printf("The element deleted = %d\n",Q[front]);
                        front=(front+1)%max;
                    }break;
            case 3 : if(front==-1){
                        printf("Queue is empty\nredirecting to main menu...\n");
                    }else{
                        printf("The current elememts in the circular Queue\n");
                        if(front<=rear){
                            for(i=front;i<=rear;i++){
                                printf("%d\n",Q[i]);
                            }
                        }else{
                            for(i=front;i<max;i++){
                                printf("%d\n",Q[i]);
                            }
                            for(i=0;i<=rear;i++){
                                printf("%d\n",Q[i]);
                            }
                        }
                        printf("\n");
                    }break;
            case 4 : printf("exiting the program...\n");
                    exit(0);
            default: printf("Something went wrong !!!\nterminating the program...\n");
                    exit(0);
                }
        }
        printf("\n");
}
```