POORNIMA

EXPERIMENT ⇒ 3

★ Objective ⇒ Identify whether given string is keyword or not.

★ Program ⇒

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

void main(){
    char a[5][10]={"printf", "scanf",
                   "if", "else", "break"};
    char str[10];
    int i, flag;
    clrscr();

    puts("Enter the string::");
    gets(str);

    for(i=0; i<strlen(str); i++){
        if(strcmp(str, a[i]==0)){
```

```
if (flag==1)
   puts ("Keyword");
else
   puts ("String");

getch();
```

3

Cut
09/09/24

POORNIMA

EXPERIMENT ⇒ 4

★ Objective ⇒ Count total no of keywords in a file.

★ Program ⇒

```
# include < stdio.h >
# include < stdlib.h >
# include < string.h >
# include < ctype.h >

Static int count = 0;
int isKeyword (char buffer[]){
    char keywords[32][10] = {"auto", "break",
                             "case", "char", "const",
"continue", "default", "do", "double", "float", "if",
"else", "void", "union", "while"};

    int i, flag = 0;

    for (i=0; i<32; ++i){
        if (strcmp (keyword[i], buffer)==0){
            flag = 1;
            count ++;
        }
    }
    return flag;
}
```

O/P ⇒ Break case
2

POORNIMA

```
int main () {
    char ch, buffer[15];
    FILE * fp;
    int i, j = 0

    fp = fopen("may.C", "r");
    if( fp == NULL ) {
        printf("error opening file\n");
        exit (0);
    }

    while((ch = fgetc(fp)) != EOF) {
        if (isalnum(ch)) {
            buffer[ j++ ] = ch;
        }
        else if((ch == ' ' || ch == '\n') && (j != 0)) {
            buffer[j] = '\0';
            j = 0;

            if (iskeyword (buffer) == 1)
                printf("%s is keyword\n", buffer);
        }
    }
    printf("no of keywords = %d", count);
    fclose (fp);
    return 0;
}
```

Page No.....

Input :- Mayur · c ·

★ O/P =>

Op → +
−
*

\+ − *

3

Total No. of Operator = 3

POORNIMA

EXPERIMENT ⇒ 5

★ Aim ⇒ Count total no of operators in a file

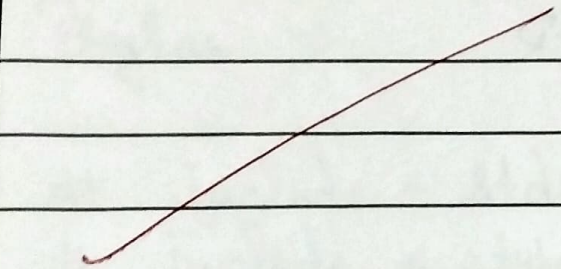
★ Program ⇒

```
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
static int count = 0;

int main(){
    char ch, buffer[15], operators[] =
    "+-*/%=";  FILE *fp;
    int i; clrscr();
    fp = fopen("KESHAV.C", "r");
    if(fp==NULL){
        printf("error while opening file\n");
        exit(0);
    }
while((ch=fgetc(fp))!=EOF){
    for(i=0; i<6; ++i){
        if(ch==operators[i]){
            printf("%c is operator\n", ch);
            count++;
        }
    }
}
```

Mayur

```
printf("no of operators = %d", count);
fclose(fp);
return 0;
}
```



Out

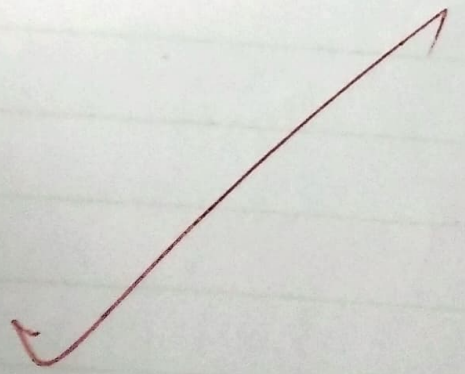
02/10/24

O/P =>

Enter no of productions : 3
Enter production 1 : E=TR
Enter production 2 : R=+TR
Enter production 3 : T=a

First (E) = {a}
First (R) = {+}
First (T) = {a}

Follow (E) = {$}
Follow (R) = {$}
Follow (T) = {$+}



POORNIMA

EXPERIMENT ⇒ 6

★ Aim ⇒ Program to implement the first and follow function by using C language.

★ Program ⇒

```
#include <stdio.h>
#include <ctype.h>
#include <string.h>
#define MAX 10

void findFirst(char, int, int);
void findFollow(char);
void addToResultSet(char, char[]);
int count, n = 0;
char production[MAX][MAX], firstSet[MAX],
  followSet[MAX];

int main(){
  printf("Enter no of productions : ");
  scanf("%d", &count);
  for(int i=0; i<count; i++){
    printf("Enter production %d:", i+1);
    scanf("%s", production[i]);
  }
```

```
for (int i=0; i<count; i++){
    findFirst(production[i][0], 0, 0);
    printf("First(%c)={", production[i][0]);
    for(int j=0; firstSet[j]!='\0'; j++)
        printf("%c", firstSet[j]);
    printf("}\n");
}

printf("\n");
for(int i=0; i<count; i++){
    findFollow(production[i][0]);
    print("Follow(%c)={", production[i][0]);
    for(int j=0; followSet[j]!='\0'; j++)
        printf("%c", followSet[j]);
    printf("}\n");
}
return 0;
}

void findFirst(char c, int q1, int q2){
    if(!isupper(c)){
        firstSet[q1]=c;
    else{
        for(int j=0; j<count; j++){
            if(production[j][0]==c){
                if(production[j][2]=='#'){
                    firstSet[q1]='#';
```

POORNIMA

```
            else {
                findFirst(production[j][2], q1, q2+1);
            }
        }
    }
}

void findFollow(char c){
    if(production[0][0]==c) addToResultSet('$', followSet);
    for(int i=0; i<count; i++){
        for(int j=2; production[i][j]!='\0'; j++){
            if(production[i][j+1]!='\0'){
                findFirst(production[i][j+1], 0, 0);
                for(int k=0; firstSet[k]!='\0'; k++)
    if(production[i][j+1]=='\0' && c!=production[i][0]){
        for(int k=0; followSet[k]!='\0'; k++)
            addToResultSet(followSet[k], followSet);
            }
        }
    }
}

void addToResultSet(char c, char set[]){
    for(int i=0; set[i]!='\0'; i++)
        if(set[i]==c) return;
    set[strlen(set)]=c;
    set[strlen(set)+1]='\0';
}
```

Page No.....