

SSN COLLEGE OF ENGINEERING (Autonomous)

Affiliated to Anna University

DEPARTMENT OF CSE

UCS 1211 PROGRAMMING IN C LABORATORY

Assignment 5

Pointer in C

Reg Number : 185001131

Name : Sai Charan B

Class : CSE – B

1. Write a C function that searches a given word in a line of text and returns the frequency count. Make use of pointer notation.

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

void frequency(char *str, char *word)
{
    int c=0,i,j,flag;
    for(i=0; *(str+i)!='\0'; i++)
    {
        if(*(str+i) == *(word+0))
        {
            flag = 1;
            for(j=0;j<strlen(word);j++)
                if(*(str+i+j) != *(word+j))
            {flag=0;
                break;
            }

            if(flag==1)
                c+=1;
        }
    }
}
```

```
        }
    }
    printf("\nFrequency of the word is: %d",c);
}
void main()
{
    char *str[50],*word[20];
    printf("\nEnter string: ");
    scanf(" %[^\\n]", str);
    printf("\nEnter word to be searched: ");
    scanf(" %[^\\n]", word);
    frequency(str, word);
}
```

Output:

cseb131@jtl-29:~\$./frequency

Enter string: abcdefcatcat

Enter word to be searched: cat

Frequency of the word is: 2

2. Given multiple lines of text, parse the text to separate the tokens. A token is a word separated by a space. Store the multiple lines of text as individual strings whose maximum length is unspecified. Maintain a pointer to each string within a one-dimensional array of pointers. Identify the last line of text in some pre-determined manner.

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

void main()
{
    char *str[30];
    int n = 0, i, j;
    for(i=0; i<30; i++)
        str[i] = (char *)malloc(sizeof(char)*30);

    do
    {
        printf("\nEnter text.Enter \"STOP\" to stop read.\n");
        scanf(" %[^\\n]s", str[n]);
    }while(strcmp(str[n++], "STOP")!=0);
    n--;
    printf("\nThe Tokens are: ");
    for(i=0;i<n;i++)
    {
        printf("\nString %d:\n", i+1);
        for(j=0;*(str[i]+j);j++)
        {
            if(*(str[i]+j)!= ' ')
                printf("%c", *(str[i]+j));
        }
    }
}
```

```
                else
                    printf("\n");
            }
        }
    }
```

Output:

cseb131@jtl-29:~\$./token

Enter text.Enter "STOP" to stop read.

i am from ssn

Enter text.Enter "STOP" to stop read.

from cse b

Enter text.Enter "STOP" to stop read.

STOP

The Tokens are:

String 1:

i

am

from

ssn

String 2:

from

cse

b

3. Implement Example program 11.22 (Adding two tables of numbers) (Page 11.26) of text book (Byron Gottfried). Modify the program creating another version so that each element in the table c is the larger of the corresponding elements in tables a and b (rather than the sum of the corresponding elements in a and b).

```
#include<stdio.h>
#include<stdlib.h>
void add(int **A,int **B,int **C,int r,int c)
{
    int i,j;
    for(i=0;i<r;i++)
        for(j=0;j<c;j++)
            * (* (C+i) +j) = * (* (A+i) +j) + * (* (B+i) +j) ;
}

void read(int **A,int r,int c)
{
    int i,j;
    for(i=0;i<r;i++)
        for(j=0;j<c;j++)
            scanf("%d",&* (A+i) +j) ;
}

void print(int **A,int r,int c)
{
    int i,j;
```

```

        for(i=0;i<r;i++)
        {
            printf("\n");
            for(j=0;j<c;j++)
                printf("%d ",*(*(A+i)+j));

        }
    }

void main()
{

    int *A[20],*B[20],*C[20],r,c,i;
    printf("Enter number of rows and columns");
    scanf("%d %d",&r,&c);
    for(i=0;i<r;i++)
    {
        A[i]=(int *)malloc(c*sizeof(int));
        B[i]=(int *)malloc(c*sizeof(int));
        C[i]=(int *)malloc(c*sizeof(int));
    }
    printf("\nEnter Table A data\n");
    read(A,r,c);
    printf("\nEnter Table B data\n");
    read(B,r,c);

    add(A,B,C,r,c);
    printf("\nTable C:");
    print(C,r,c);
}

```

Output:

```
cseb131@jtl-29:~$ ./matrixadd
```

```
Enter number of rows and columns3 3
```

```
Enter Table A data
```

```
1
```

```
3
```

```
2
```

```
5
```

```
7
```

```
8
```

```
4
```

```
5
```

```
3
```

```
Enter Table B data
```

```
8
```

```
5
```

```
3
```

```
6
```

```
7
```

```
3
```

```
7
```

```
5
```

```
6
```

```
Table C:
```

```
9 8 5
```

```
11 14 11
```

```
11 10 9
```

Larger (modification of 3)

Only change in add function

```
void maximum(int **A,int **B,int **C,int r,int c)
{
    int i,j;
    for(i=0;i<r;i++)
        for(j=0;j<c;j++)
            if(*(*(A+i)+j)>*(*(B+i)+j))
                *(*(C+i)+j)=*(*(A+i)+j);
            else
                *(*(C+i)+j)=*(*(B+i)+j);
}
```

Output:

```
cseb131@jtl-29:~$ ./matrixmax
```

```
Enter number of rows and columns3
```

```
3
```

```
Enter Table A data
```

```
1
```

```
2
```

```
3
```

```
4
```

```
5
```

```
6
```

```
7
```

```
8
```

```
9
```


Enter Table B data

9

8

7

6

5

4

3

2

1

Table C:

9 8 7

6 5 6

7 8 9

4. Modify the program shown in Example 11.28 (displaying the day of the year) (Page 11.37) of text book (Byron Gottfried) so that it can determine the number of days between two dates, assuming both dates are beyond the base date of January 1, 1900

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

void calendar(int *d, int *m, int *y)
{
    scanf(" %d %d %d", d,m,y);
}

void main()
{
    printf("\nEnter from date\n");
    calendar(&d1, &m1, &y1);
    printf("\nEnter to date\n");
    calendar(&d2, &m2, &y2);
}
```