

```
#if 0

/***** Exercise 3  -17 *****/
#include<stdio.h>
#define MAXLINE 1000
#define LIMIT 8

/* we call it ngetline, for new getline so that it does not conflict with system function getline */
int ngetline(char line[],int lim);

int main(void)
{
    int len;
    char line[MAXLINE];

    while((len=nggetline(line,MAXLINE)) > 0)
    {
        if( len > LIMIT )
            printf("%s",line);
    }

    return 0;
}

int ngetline(char s[],int lim)
{
    int i,c;

    for (i = 0; i < lim - 1 && (c = getchar ()) != EOF && c != '\n' ; ++i)
        s[i]=c;

    if(c == '\n')
    {
        s[i] = c;
        ++i;
    }
    s[i] = '\0';

    return i;
}

#endif

#if 0

/***** calculator using  function pointer *****/

#include<stdio.h>
void add()
{
    printf("add fun\n");
    return;
}
void sub()
{
    printf("sub fun\n");
    return;
}

int main(void)
{
    int num;
    void (*fun[])() = { add,sub };

    printf("enter the number 0)add      1)sub\n");
    fgets((char*)num,100,stdin);

    num = atoi((char*)&num);
    if(num>=0 && num<=1)
    {
        (*fun[num])();
    }
}
```

```
    }

    return 0;
}

#endif

#if 0
/***** remove duplicate char in a string *****/

#include<stdio.h>
int main(void)
{
    char a[] = "hellohdbvgstwr";
    int i,j;
    for(i=0 ; a[i];i++)
    {
        for(j=0;a[j];j++)
        {
            if(a[i] == a[j])
                break;
        }
        if(i==j)
            printf("%c ",a[i]);
    }
    return 0;
}

#endif

#if 0
/***** print duplicate char *****/
#include<stdio.h>
#include<string.h>

void print_dup(char* str)
{
    int len = strlen(str);
    int a[100],i;
    char temp[len];

    for(i=0;i<len;i++)
    {
        int t = (int)str[i] - 65;
        if(a[t]!=1)
        {
            printf("%c",str[i]);
        }
        a[t] = 1;
    }
}

int main()
{
    print_dup("BANANAS");
    return 0;
}

#endif

#if 0
/***** largest sum in array elements *****/
#include<stdio.h>
int main(void)
{
```

```

int a[10] = {-1,-30,-3,-4,5,6,-7,8,-9,1};
int i,j;
int temp,lar_sum=0;
for(i = 0; i < 9; i++)
{
    temp = a[i] + a[i+1] ;

    if ( temp > lar_sum )
    {
        lar_sum = temp;
    }
}
printf("%d\n",lar_sum);
return 0;
}

#endif

#if 0
/***** largest element using bitwise *****/

#include<stdio.h>
int main(void)
{
    int a[10] = {1,30,3,4,50,6,7,8,9,10};
    int i,j;
    int temp,lar=0;
    for(i = 0; i < 10; i++)
    {
        temp = a[i] - lar;
        lar = ((temp >> 31 & 1) ? ( lar ) : ( a[i] ));
    }
    printf("%d\n",lar);
    return 0;
}

#endif

#if 0

/***** print matrix in spiral form *****/

#include <stdio.h>
#define R 4
#define C 4

void spiralPrint(int m, int n, int a[R][C])
{
    int i, k = 0, l = 0;

    /* k - starting row index
       m - ending row index
       l - starting column index
       n - ending column index
       i - iterator
    */

    while (k < m && l < n)
    {
        /* Print the first row from the remaining rows */
        for (i = l; i < n; ++i)
        {
            printf("%d ", a[k][i]);
        }
        k++;

        /* Print the last column from the remaining columns */
        for (i = k; i < m; ++i)
        {
            printf("%d ", a[i][n-1]);
        }
        n--;
    }
}

```

```

        /* Print the last row from the remaining rows */
        if ( k < m)
        {
            for (i = n-1; i >= l; --i)
            {
                printf("%d ", a[m-1][i]);
            }
            m--;
        }

        /* Print the first column from the remaining columns */
        if (l < n)
        {
            for (i = m-1; i >= k; --i)
            {
                printf("%d ", a[i][l]);
            }
            l++;
        }
    }
}

/* Driver program to test above functions */
int main()
{
    int a[R][C] = { 1,  2,  3,  4,  5,  6,
                    7,  8,  9, 10, 11, 12,
                    13, 14, 15, 16
    };

    spiralPrint(R, C, a);
    return 0;
}

#endif

#if 0

/***** memmove implimentation *****/
#include<stdio.h>
int main(void)
{
    char src[] = "sdde5";
    char dst[] = "hello";
    int n = 3,i;
    char *d = dst;
    char *s = src;

    if(dst == src)
    {
        printf("%s\n",dst);
    }
    if(dst < src)
    {
        for(i=0;i<3;i++)
            dst[i] = src[i];
        printf("%s\n",dst);
    }
    else
    {
        while(n--)
        {
            d[i] = s[i];
        }
        printf("%s",d);
    }
    return 0;
}

```

```
#endif
```

```
#if 0
/***** multiplication using recursion *****/
#include<stdio.h>
```

```
int muli(int n1,int n2)
{
    if(n2 == 1)
        return n1;
    else
    {
        return n1+muli(n1,(n2-1));
    }
}
```

```
int main(void)
{
    int n1 =10;
    int n2 =20;

    int mul = muli(n1,n2 );
    printf("mul = %d\n",mul);
    return 0;
}
```

```
#endif
```

```
#if 0
```

```
/** remove the duplicate array elements */
#include<stdio.h>
void duplicate(int array[], int num)
{
    int *count = (int *)malloc(100);
    int i;
    printf("duplicate elements present in the given array are ");
    for (i = 0; i < num; i++)
    {
        if (count[array[i]] == 1)
            printf(" %d ", array[i]);
        else
            count[array[i]]++;
    }
}
```

```
int main()
{
    int array[] = {5, 10, 10, 2, 1, 4, 2};
    int array_freq = sizeof(array) / sizeof(array[0]);
    duplicate(array, array_freq);
    return 0;
}
```

```
/*
```

```
int main(void)
{
    int a[] = {1,2,3,2,3,4,5,4,3,2,3,4,5};
    int n = sizeof(a) / sizeof(a[0]);
    int i,j;
    for(i=0;i<n;i++)
    {
        for(j=0;j<n;j++)
        {
            if(a[i] == a[j])
                break;
        }
    }
}
```

```

        }
        if(i==j)
        {
            printf("%d ",a[i]);
        }
    }
    return 0;
}

*/

#endif

#if 0
/***** print with out printf function *****/
#include<stdlib.h>
#include<string.h>
#include<stdio.h>
int main()
{
    write(1,"hello world",strlen("hello world"));

    return 0;
}
#endif

#if 1

/***** remove sub string from main string *****/

#include <stdio.h>
#include <string.h>
void main()
{
    int i, j = 0, k = 0, count = 0;
    char str[100], key[20];
    char str1[10][20];
    printf("enter string:");
    scanf("%[^\n]s",str);
    /* Converts the string into 2D array */
    for (i = 0; str[i] != '\0'; i++)
    {
        if (str[i]==' ')
        {
            str1[k][j] = '\0';
            k++;
            j = 0;
        }
        else
        {
            str1[k][j] = str[i];
            j++;
        }
    }
    str1[k][j] = '\0';
    printf("enter key:");
    scanf("%s", key);
    /* Compares the string with given word */
    for (i = 0; i < k + 1; i++)
    {
        if (strcmp(str1[i], key) == 0)
        {
            for (j = i; j < k + 1; j++)
                strcpy(str1[j], str1[j + 1]);
            k--;
        }
    }
    for (i = 0; i < k + 1; i++)
    {
        printf("%s ", str1[i]);
    }
}

```

```
    }
}

#endif

#if 0
/***** reverse the words *****/
#include<stdio.h>

void reverseWords( char * str )
{
    int i = 0, j = 0;
    reverseString( str, strlen(str) );
    printf("%s\n",str);

    while( 1 )
    {
        if( str[j] == ' ' )
        {
            reverseString( str+i, j-i );
            i = j+1;
        }
        if( str[j] == '\\0')
        {
            break;
        }
        j++;
    }
}

void reverseString(char* str, int len)
{
    int i, j;
    char temp;
    i=j=temp=0;

    j=len-1;
    for (i=0; i<j; i++, j--)
    {
        temp=str[i];
        str[i]=str[j];
        str[j]=temp;
    }
}

int main(void)
{
    char a[100] = "i am fine";

    reverseWords(a);
    printf("%s\n",a);
    return 0;
}

#endif

#if 0
/***** insert the string in a given string *****/
#include<stdio.h>
#include<string.h>
void main()
{
    char str1[20], str2[20];
    int l1, l2, n, i;
    puts("Enter the string 1\n");
    scanf("%s",str1);
```

```
    l1 = strlen(str1);
    puts("Enter the string 2\n");
    scanf(" %s",str2);

    l2 = strlen(str2);
    printf("Enter the position where the string is to be inserted\n");
    scanf(" %d", &n);
    for(i = n; i < l1; i++)
    {
        str1[i + l2] = str1[i];
    }
    for(i = 0; i < l2; i++)
    {
        str1[n + i] = str2[i];
    }
    str2[l2 + 1] = '\0';
    printf("After inserting the string is %s", str1);
}

#endif
```