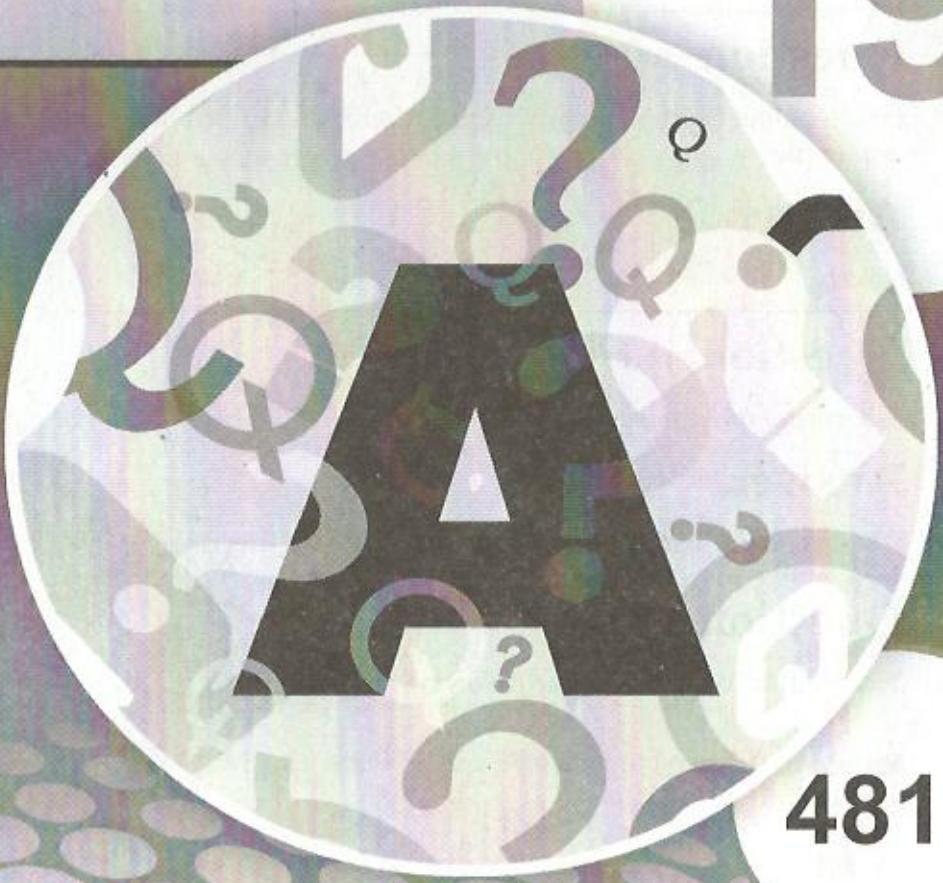
Library Functions

19



481

Question 19.1

Is standard library a part of C language?

Answer

No.

Question 19.2

Input / output function prototypes and macros are defined in which header file?

Answer

stdio.h

Question 19.3

What are *stdin*, *stdout* and *stderr*?

Answer

Standard input, standard output and standard error streams.

Question 19.4

What will be the output of the following program?

```
#include <stdio.h>
int main( )
{
    int i ;
    i = printf ( "How r u ?\n" ) ;
    i = printf ( "%d\n", i ) ;
    printf ( "%d\n", i ) ;
    return 0 ;
}
```


Answer

```
How r u ?
10
3
```

Question 19.5

What will be the output of the following program?

```
#include <stdio.h>
int main( )
{
    int i ;
    i = scanf ( "%d %d", &i, &i ) ;
    printf ( "%d\n", i ) ;
    return 0 ;
}
```

Answer

2

Question 19.6

What will be the output of the following program?

```
#include <stdio.h>
int main( )
{
    char str[ ] = "Zingle Bell Zingle Bell" ;
    printf ( "%.#s %.2s\n", str, str );
    return 0 ;
```

```
}
```

Answer

#s Zi

Question 19.7

Point out error, if any, in the following program.

```
#include <stdio.h>
int main( )
{
    fprintf ( ".#MadhurAmla" ) ;
    printf ( "%.ef\n", 2.0 ) ;
    return 0 ;
}
```

Answer

Error: *'fprintf'* function does not take 1 parameters.

Question 19.8

What will be the output of the following program?

```
#include <stdio.h>
int main( )
{
    int i ;
    char c ;
    for ( i = 1 ; i <= 5 ; i++ )
    {
        scanf ( "%c", &c ) ;
        printf ( "%c", c ) ;
        ungetc ( c, stdin ) ;
    }
    return 0 ;
}
```


Answer

You will get a chance to enter the character only once. Whichever character you enter will be printed five times.

Question 19.9

What will be the output of the following program?

```
#include <stdio.h>
int main( )
{
    int i ;
    char c ;
    for ( i = 1 ; i <= 5 ; i++ )
    {
        scanf ( "%c", &c ) ;
        ungetc ( c, stdout ) ;
        printf ( "%c", c ) ;
        ungetc ( c, stdin ) ;
    }
    printf ( "\n" ) ;
    return 0 ;
}
```

Answer

The character that is supplied to *scanf( )* gets printed 5 times.

Question *19.10*

What is the difference between the functions *rewind( )* and *fsetpos( )*?

Answer

rewind( ) function repositions file pointer to streams beginning. *fsetpos( )* positions the file pointer of a stream at desired position. Position is obtained using *getpos( )* function.

Question *19.11*

What is the purpose of *flushall( )* and *fflush( )* function?

Answer

(a) *flushall( )* clears all buffers associated with open input streams, and writes all buffers associated with open output streams to their respective files.

(b) *fflush( )* flushes a specified stream. *fflush (NULL)* flushes all streams.

Question *19.12*

What will be the output of the following program?

```
#include <stdio.h>
#include <math.h>
int main( )
{
    float i = 2.5 ;
    printf ( "%f %d\n", floor ( i ), ceil ( i ) ) ;
    return 0 ;
}
```


Answer

2.000000 0

Question *19.13*

What do the functions *atoi( )*, *itoa( )* and *gcvt( )* do? Show how you will use them in a program?

Answer

atoi( ) Converts a string to an integer.
itoa( ) Converts an integer to a string.
gcvt( ) Converts a floating-point number to a string.

```
#include <stdio.h>
#include <stdlib.h>
int main( )
{
    char s[ ] = "12345" ;
    char buffer[15], string[20] ;
    int i ;
    i = atoi ( s ) ;
    printf ( "%d\n", i ) ;
    gcvt ( 20.141672, 4, buffer ) ;
    printf ( "%s\n", buffer ) ;
    itoa ( 15, string, 2 ) ;
    printf ( "%s\n", string ) ;
    return 0 ;
}
```

Question *19.14*

What will be the output of the following program?

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

```
#include <stdlib.h>
int main( )
{
    char *i = "55.555" ;
    int result1 = 10 ;
    float result2 = 11.111 ;
    result1 = result1 + atoi ( i ) ;
    result2 = result2 + atof ( i ) ;
    printf ( "%d %f\n", result1, result2 ) ;
    return 0 ;
}
```

Answer

65 66.666000

Question *19.15*

Does there exist any other function, which can be used to convert an *int* or a *float* to a string? If yes, show how you will use it?

Answer

The function *sprintf( )* can be used for this purpose. This function also has the ability to format the numbers as they are converted to strings. The following program shows how to use this function.

```
#include <stdio.h>
int main( )
{
    int a = 25 ;
    float b = 3.14 ;
    char str[40] ;
    sprintf ( str, "a = %d b = %f\n", a, b ) ;
    puts ( str ) ;
    return 0 ;
}
```


Question *19.16*

How will you use *qsort( )* function to sort an array of structures?

Answer

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
struct stud
{
    int rollno ;
    int marks ;
    char name[30] ;
} ;
int sort_rn ( const void *, const void * ) ;
int sort_name ( const void *, const void * ) ;
int sort_marks ( const void *, const void * ) ;

int main( )
{
    static struct stud ss[ ] = {
                                { 15, 96, "Akshay" },
                                { 2, 97, "Madhuri" },
                                { 8, 85, "Aishvarya" },
                                { 10, 80, "Sushmita" }
                              } ;
    int x, w ;
    w = sizeof ( struct stud ) ;

    printf ( "\n\nIn order of roll numbers: " ) ;
    qsort ( ss, 4, w, sort_rn ) ;
    for ( x = 0 ; x < 4 ; x++ )
        printf ( "%d %s %d\n", ss[x].rollno, ss[x].name, ss[x].marks ) ;

    printf ( "\n\nIn order of names: " ) ;
```

```
    qsort ( ss, 4, w, sort_name ) ;
    for ( x = 0 ; x < 4 ; x++ )
        printf ( "%d %s %d\n", ss[x].rollno, ss[x].name, ss[x].marks ) ;

    printf ( "\n\nIn order of marks: " ) ;
    qsort ( ss, 4, w, sort_marks ) ;
    for ( x = 0 ; x < 4 ; x++ )
        printf ( "%d %s %d\n", ss[x].rollno, ss[x].name, ss[x].marks ) ;
    return 0 ;
}

int sort_rn ( const void *a, const void *b )
{
    struct stud *t1 = ( struct stud * ) a ;
    struct stud *t2 = ( struct stud * ) b ;
    return ( t1->rollno - t2->rollno ) ;
}

int sort_name ( const void *a, const void *b )
{
    struct stud *t1 = ( struct stud * ) a ;
    struct stud *t2 = ( struct stud * ) b ;
    return ( strcmp ( t1->name, t2->name ) ) ;
}

int sort_marks ( const void *a, const void *b )
{
    struct stud *t1 = ( struct stud * ) a ;
    struct stud *t2 = ( struct stud * ) b ;
    return ( t2->marks - t1->marks ) ;
}
```

Question *19.17*

How will you use *qsort( )* function to sort the names stored in an array of pointers to strings?

Answer

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

int sort_name ( const void *, const void * ) ;
int main( )
{
    char *names[ ] = {
                        "Akshay",
                        "Madhuri",
                        "Aishvarya",
                        "Sushmita",
                        "Sudeepta"
                    } ;
    int i ;
    qsort ( names, 5, sizeof ( char * ), sort_name ) ;
    for ( i = 0 ; i < 5 ; i++ )
        printf ( "%s\n", names[i] ) ;
    return 0 ;
}

int sort_name ( const void *t1, const void *t2 )
{
    /* t1 and t2 are always pointers to objects being compared */
    char **t11, **t22 ;
    /* cast appropriately */
    t11 = ( char ** ) t1 ;
    t22 = ( char ** ) t2 ;
    return ( strcmp ( *t11, *t22 ) ) ;
}
```

Question 19.18

How will you use *bsearch( )* function to search a name stored in an array of pointers to strings?

Answer

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
int sort_name ( const void *, const void * ) ;
int bs_compare ( const void *, const void * ) ;

int main( )
{
    char *names[ ] = {
                        "Akshay",
                        "Madhuri",
                        "Aishvarya",
                        "Sushmita",
                        "Sudeepta"
                    } ;
    int i, width, num_ele ;
    char *s = "Madhuri", **b ;

    qsort ( names, 5, sizeof ( char * ), sort_name ) ;

    for ( i = 0 ; i < 5 ; i++ )
        printf ( "\n%s", names[ i ] ) ;

    width = sizeof ( names[ 0 ] ) ;
    num_ele  = sizeof ( names ) / width ;
    b = ( char** ) bsearch ( &s, names, num_ele, width, bs_compare ) ;

    if ( b == NULL )
        printf ( "Not found\n" ) ;
    else
        printf ( "Found %s\n", *b ) ;
}

int sort_name ( const void *t1, const void *t2 )
{
    /* t1 and t2 are always pointers to objects being compared */
    char **t11, **t22 ;
    /* cast appropriately */
    t11 = ( char ** ) t1 ;
    t22 = ( char ** ) t2 ;
    return ( strcmp ( *t11, *t22 ) ) ;
    return 0 ;
}

int bs_compare ( const void *a, const void *b )
{
    char** s1 = ( char** ) a ;
    char** s2 = ( char** ) b ;
    return ( strcmp ( *s1, *s2 ) ) ;
}
```


Question 19.19

How will you use the functions *sin( ), pow( ), sqrt( )*?

Answer

```
#include <stdio.h>
#include <math.h>
int main( )
{
    int ang ;
    float angrad, x, y, a, b ;
    printf ( "Enter the angle in degrees\n" ) ;
    scanf ( "%d", &ang ) ;
    angrad = ang * 3.14 / 180 ;
    x = sin ( angrad ) ;
```

```
    a = pow ( x, 2 ) ;
    b = 1 - a ;
    y = sqrt ( b ) ;
    printf ( "cosine of angle %d is = %f\n", ang, y ) ;
    return 0 ;
}
```

Question *19.20*

How will you use the function *memcpy( )*?

Answer

```
#include <stdio.h>
#include <memory.h>
#include <malloc.h>
int main( )
{
    int area ;
    char src[ ] = "Pray, not for lighter load, but for stronger back" ;
    char *dest ;
    area = sizeof ( src ) ;
    dest = ( char * ) malloc ( area ) ;
    memcpy ( dest, src, area ) ;
    printf ( "%s\n", src ) ;
    printf ( "%s\n", dest ) ;
    return 0 ;
}
```

Question *19.21*

What will be the output of the following program?

```
#include <stdio.h>
#include <string.h>
int main( )
{
    char str1[ ] = "Master of U\0niverse", str2[20] ;
    char *p ;
    p = ( char* ) memccpy ( str2, str1, 'i', 20 ) ;
    *p = '\0' ;
    printf ( "%s\n", str2) ;
    return 0 ;
}
```

Answer

Master of U

Question *19.22*

How will you use the function *memset( )*?

Answer

```
#include <stdio.h>
#include <string.h>
#include <mem.h>  /* delete this while using Visual Studio */
int main( )
{
    int  area ;
    char src[ ] = "Bon jour, Madam" ;
    area = sizeof ( src ) ;
    memset ( src, '!', area - 7 ) ;
    printf ( "%s\n", src ) ;
    return 0 ;
}
```

Question *19.23*

How will you use the function *memmove( )*?

Answer

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main( )
{
    int area ;
    char *dest ;
    char src[ ] = "Life is a camera and you are the target "
                  "so keep smiling. Always!" ;
    area = sizeof ( src ) ;
    dest = ( char * ) malloc ( area ) ;
    memmove ( dest, src, area ) ;
    printf ( "%s\n", dest ) ;
    printf ( "%s\n", src ) ;
    return 0 ;
}
```

Question *19.24*

What will be the output of the following program?

```
#include <stdio.h>
#include <string.h>
int main( )
{
    char dest[ ] = { 97, 97, 0 } ;
    char src[ ] = "aaa" ;
    int i ;
    if ( ( i = memcmp ( dest, src, 2 ) ) == 0 )
        printf ( "You got it\n" ) ;
    else
        printf ( "You missed it\n" ) ;
    return 0 ;
}
```


Answer

You got it

Question *19.25*

How will you use the functions *fseek( )*, *fread( )*, *fwrite( )* and *ftell( )*?

Answer

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
struct stud
{
    int rollno ;
    char name[10] ;
    float per ;
} e ;
FILE *fs ;
int main( )
{
    long position = 0L ;
    int rollno ;
    char ch ;
    float temp ;
    fs = fopen ( "stud.dat", "rb+" ) ;
    if ( fs == NULL )
    {
        puts ( "Unable to open file" ) ;
        exit ( 1 ) ;
    }
    do
    {
        printf ( "Enter code no. to modify:\n" ) ;
        scanf ( "%d", &rollno ) ;
```

```
        while ( fread ( &e, sizeof ( e ), 1, fs ) == 1 )
        {
            if ( e.rollno == rollno )
            {
                printf ( "Enter the new record\n" ) ;
                scanf ( "%s %f", e.name, &temp ) ;
                e.per = temp ;
                fseek ( fs, position, SEEK_SET ) ;
                fwrite ( &e, sizeof ( e ), 1, fs ) ;
                break ;
            }
            position = ftell ( fs ) ;
        }
        puts ( "You want to modify records" ) ;
        ch = getche( ) ;
    } while ( ch == 'Y' ) ;
    return 0 ;
}
```

Question *19.26*

How will you obtain the current time and difference between two times?

Answer

```
#include <stdio.h>
#include <time.h>
int main( )
{
    time_t t1, t2 ;
    double diff, f ;
    int i = 2 ;
    time ( &t1 ) ;
    /* some statements here */
    time ( &t2 ) ;
    diff = difftime ( t2, t1 ) ;
```



```
    printf ( "Program was active for %lf seconds\n", diff ) ;
    return 0 ;
}
```

Question *19.27*

How will you obtain the current date and time programmatically?

Answer

```
#include <time.h>
#include <stdio.h>
int main( )
{
    char datebuf[9] ;
    char timebuf[9] ;
    _strdate ( datebuf ) ;
    _strtime ( timebuf ) ;
    printf ( "Date: %s  Time: %s\n", datebuf, timebuf ) ;
    return 0 ;
}
```

Question *19.28*

How will you generate true random numbers?

Answer

```
#include <stdio.h>
#include <time.h>
#include <stdlib.h>
int main( )
{
    int i, j ;
    srand (  time ( NULL ) ) ;  /* use randomize( ) in case of TC/TC++ */
    for ( i = 0 ; i < 10 ; i++ )
```

```
    {
        j = rand( ) ; /* use random( ) in case of TC/TC++ */
        printf ( "%d\n", j ) ;
    }
    return 0 ;
}
```

Question *19.29*

Will the following program always output 'Banglore'?

```
#include <stdio.h>
#include <string.h>
int main( )
{
    char str1[ ] = "Banglore - 440010" ;
    char str2[10] ;
    strncpy ( str2, str1, 8 ) ;
    printf ( "%s\n", str2 ) ;
    return 0 ;
}
```

Answer

No. Because after copying the source string into the target string *strncpy( )* doesnt terminate the target string with a '\0' if length of source string is greater than or equal to 8 (in this case). A better way of copying will be:

```
str2[0] = '\0' ;
strncat ( str2, str1, 8 ) ;
```

strncat( ) always terminates the target string with a '\0'.

Question *19.30*

Can you shorten this code?

```
#include <stdio.h>
#include <string.h>
int main( )
{
    char str1[ ] = "Banglore - 440010" ;
    char str2[10] ;
    str2[0] = '\0' ;
    strncat ( str2, str1, 8 ) ;
    printf ( "%s\n", str2 ) ;
    return 0 ;
}
```

Answer

Yes, using *sprintf( )* as shown below:

```
#include <stdio.h>
int main( )
{
    char str1[ ] = "Banglore - 440010" ;
    char str2[10] ;
    sprintf ( str2, "%.*s", 8, str1 ) ;
    printf ( "%s\n", str2 ) ;
    return 0 ;
}
```

Question *19.31*

How will you implement a *substr( )* function that extracts a substring from a given string?

Answer

```
#include <stdio.h>
#include <string.h>
void substr ( char *t, char *s, int pos, int len ) ;
int main( )
{
    char str1[ ] = "Banglore" ;
    char str2[5] ;
    /* extract 3 characters beginning with first character */
    substr ( str2, str1, 1, 3 ) ;
    printf ( "%s\n", str2 ) ;
    return 0 ;
}
void substr ( char *t, char *s, int pos, int len )
{
    t[0] = '\0' ;
    strncat ( t, s + pos, len ) ;
}
```

Question *19.32*

Given a sentence containing words separated by spaces how will you construct an array of pointers to strings containing addresses of each word in the sentence?

Answer

```
#include <stdio.h>
#include <string.h>
int main( )
{
    char str[ ] = "This is a test" ;
    char *ptr[10] ;
    char *p ;
    int i = 1, j ;
```



```
    p = strtok ( str, " " ) ;
    if ( p != NULL )
    {
        ptr[ 0 ] = p ;
        while ( 1 )
        {
            p = strtok ( NULL, " " ) ;
            if ( p == NULL )
                break ;
            else
            {
                ptr[ i ] = p ;
                i++ ;
            }
        }
    }
    for ( j = 0 ; j < i ; j++ )
        printf ( "%s\n", ptr[ j ] ) ;
    return 0 ;
}
```

Question *19.33*

Write the comparison function *qs_compare( )* for the following code.

```
#include <stdio.h>
#include <stdlib.h>
struct date
{
    int d, m, y ;
} ;
int qs_compare ( const void *, const void * ) ;
int main( )
{
    struct date dd[ ] = {
                        { 17, 11, 62 },
```

```
                    { 24, 8, 78 },
                    { 17, 11, 62 },
                    { 16, 12, 76 },
                    { 19, 2, 94 }
                };
    int i, w ;
    w = sizeof ( struct date ) ;
    qsort ( dd, 5, w, qs_compare ) ;
    for ( i = 0 ; i < 4 ; i++ )
        printf ( "%d %d %d\n", dd[ i ].d, dd[ i ].m, dd[ i ].y ) ;
    return 0 ;
}
```

Answer

```
int qs_compare ( const void *p1, const void *p2 )
{
    const struct date *sp1 = ( const struct date* ) p1 ;
    const struct date *sp2 = ( const struct date* ) p2 ;
    if ( sp1 -> y < sp2 -> y )
        return ( -1 ) ;
    else if ( sp1 -> y > sp2 -> y )
        return ( 1 ) ;
    else if ( sp1->m < sp2->m )
        return ( -1 ) ;
    else if ( sp1 -> m > sp2 -> m )
        return ( 1 ) ;
    else if ( sp1 -> d < sp2 -> d )
        return ( -1 ) ;
    else if ( sp1 -> d > sp2 -> d )
        return ( 1 ) ;
    else
        return ( 0 ) ;
}
```


Question *19.34*

How will you sort a linked list?

Answer

Often its easier to keep the list in order as we build it rather than sorting it later. Still if we want to sort the list then we can allocate a temporary array of pointers, fill it with pointers to the various nodes in the list, call *qsort( )* and finally rebuild the list pointers based on the sorted array.

Question *19.35*

What's the difference between the functions *rand( ), random( ), srand( )* and *randomize( )*?

Answer

rand( ) returns a random number.
random( ) returns a random number in a specified range.
srand( ) initialises a random number generator with a given seed value.
randomize( ) initialises a random number generator with a random value based on time.

randomize( ) and *random( )* functions will work with Turbo C/C++ compiler, whereas *srand( )* and *rand( )* will work with Visual Studio and gcc compiler.

Question *19.36*

Whats the difference between the function *memmove( )* and *memcpy( )*?

Answer

Both the functions copy a block of bytes from source to destination. However, if source and destination overlaps the behaviour of *memcpy( )* is undefined, whereas *memmove( )* carries out the copying correctly. *memcpy( )* is more efficient, whereas *memmove( )* is safer to use.

Question *19.37*

How will you print a string on the printer?

Answer

```
#include <stdio.h>
int main( )
{
    char str[ ] = "There willn't have been COBOL without C" ;
    fprintf ( stdprn, "%s\n", str ) ;
    return 0 ;
}
```

Question *19.38*

Can you use the function *fprintf( )* to display the output on the screen?

Answer

Yes, by replacing the usual file pointer with *stdout* as shown below:

```
fprintf ( stdout, "%s %d %f", str, i, a ) ;
```


Question *19.39*

Which standard library function will you use to find the last occurrence of a character in a string in C?

Answer

strchr( ) or *strrchr( )*

Question *19.40*

State True or False:

A. *sscanf( )* or *atoi( )* function can be used to convert a string like "432" into an integer.

B. The prototypes of all standard library string functions are declared in the file "string.h".

C. It is necessary that for the string functions to work safely the strings must be terminated with '\0'.

D. If two strings are found to be unequal then *strcmp* returns the difference between the first non-matching pair of characters.

E. The *itoa( )* function can convert an integer in decimal, octal or hexadecimal form to a string.

F. Data written into a file using *fwrite( )* can be read back using *fscanf( )*.

G. *ftell( )* returns the current position of the pointer in a file stream.

H. *FILE* is a *structure* suitably *typedef*ed in "stdio.h".

Answer

A. True
B. True
C. True
D. True
E. True
F. False
G. True
H. True