

CHAPTER EIGHTEEN

Console Input/Output

[A] What will be the output of the following programs:

```
(a) #include <stdio.h>
#include <ctype.h>
int main( )
{
    char ch ;
    ch = getchar( ) ;
    if ( islower ( ch ) )
        putchar ( toupper ( ch ) ) ;
    else
        putchar ( tolower ( ch ) ) ;
    return 0 ;
}
```

Input *Output:*

a	A
Z	z

```
(b) #include <stdio.h>
int main( )
{
    int i = 2 ;
    float f = 2.5367 ;
    char str[ ] = "Life is like that" ;
```

```
printf ( "%4d\\%3.3f\\%4s\\n", i, f, str );
return 0 ;
}
```

Output:

2 2.537 Life is like that

```
(c) #include <stdio.h>
int main( )
{
    printf ( "More often than \\b\\b not \\rthe person who \\
            wins is the one who thinks he can!\\n" );
    return 0 ;
}
```

Output:

the person who

wins is the one who thinks he can!

```
(d) #include <conio.h>
char p[ ] = "The sixth sick sheikh's sixth ship is sick" ;
int main( )
{
    int i = 0 ;
    while ( p[ i ] != '\\0' )
    {
        putchar ( p[ i ] );
        i++ ;
    }
    return 0 ;
}
```

Output:

The sixth sick sheikh's sixth ship is sick

6) Point out the errors, if any, in the following programs:

```
#include <stdio.h>
int main( )
{
    int i;
    char a[ ] = "Hello";
    while ( a != '\0' )
    {
        printf ( "%c", *a );
        a++;
    }
    return 0;
}
```

Error. Lvalue required. Post-fix increment operator cannot be used with the name of the array.

```
#include <stdio.h>
int main( )
{
    double dval;
    scanf ( "%f", &dval );
    printf ( "Double Value = %lf\n", dval );
    return 0;
}
```

No Error. But the format specifier used for **double** in **scanf()** should be **lf** instead of **f**.

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


Error. Undefined symbol **n**.

```
(d) #include <stdio.h>
int main( )
{
    char *mess[ 5 ];
    int i;
    for ( i = 0 ; i < 5 ; i++ )
        scanf ( "%s", mess[ i ] );
    return 0 ;
}
```

Error. **i** undefined. Moreover when we use an array of pointers to strings the array should be initialised where it is declared.

```
(e) #include <stdio.h>
int main( )
{
    int dd, mm, yy ;
    printf ( "Enter day, month and year\n" );
    scanf ( "%d%*c%d%*c%d", &dd, &mm, &yy );
    printf ( "The date is: %d - %d - %d\n", dd, mm, yy );
    return 0 ;
}
```

No Error. The asterisk (*) acts as a suppression character. Giving '*' in **scanf()** would skip the assignment of next scanned value to the specified address..

```
(f) #include <stdio.h>
int main( )
{
    char text ;
    sprintf ( text, "%4d\t%2.2f\n%s", 12, 3.452, "Merry Go Round" );
    printf ( "%s\n", text );
    return 0 ;
}
```

}

No error.

(g) #include <stdio.h>
int main()

```
{  
    char buffer[ 50 ];  
    int no = 97;  
    double val = 2.34174 ;  
    char name[ 10 ] = "Shweta" ;  
  
    sprintf ( buffer, "%d %lf %s", no, val, name ) ;  
    printf ( "%s\n", buffer ) ;  
    sscanf ( buffer, "%4d %2.2lf %s", &no, &val, name ) ;  
    printf ( "%s\n", buffer ) ;  
    printf ( "%d %lf %s\n", no, val, name ) ;  
    return 0 ;  
}
```

No error.

[C] Answer the following:

(a) To receive the string "We have got the guts, you get the glory!!" in an array **char str[100]** which of the following functions would you use?

1. scanf ("%s", str) ;
2. gets (str) ;
3. getche (str) ;
4. fgetchar (str) ;

Answer:

2. gets()

(b) Which function would you use if a single key were to be received through the keyboard?

1. scanf()
2. gets()
3. getche()
4. getchar()

Answer:

3. getche()

(c) If an integer is to be entered through the keyboard, which function would you use?

1. scanf()
2. gets()
3. getche()
4. getchar()

Answer:

1. scanf()

(d) What is the difference between **getchar()**, **fgetchar()**, **getch()** and **getche()**?

Answer:

All receive a character from keyboard. There are minor differences in them:

getchar(): Receives a character from keyboard, but it is necessary to hit the enter key after the character.

fgetchar(): Same as **getchar()**. **getchar()** is a macro, whereas, **fgetchar()** is a function.

getch(): Receives a character from keyboard without echoing it on the screen

getche(): Receives a character from keyboard and echoes it on the screen

(e) Which of the following can a format string of a **printf()** function contain:

1. Characters, format specifications and escape sequences
2. Character, integers and floats
3. Strings, integers and escape sequences
4. Inverted commas, percentage sign and backslash character

Answer:

1. Characters, conversion specifications and escape sequences
- (f) The purpose of the field-width specifier in a **printf()** function is to:
1. Control the margins of the program listing
 2. Specify the maximum value of a number
 3. Control the size of font used to print numbers
 4. Specify how many columns would be used to print the number

Answer:

4. Specifies how many columns will be used to print the number

(D) Answer the following:

- (a) Define two functions **xgets()** and **xputs()** which work similar to the standard library functions **gets()** and **puts()**.

Program:

/* Function xgets() and xputs() */

include <stdio.h>

void xputs (char *);

void xgets (char *);

int main()

{

char sent[100];

xputs ("Enter a sentence ... ");

xgets (sent);

printf ("\n\n");

xputs (sent);

return 0 ;

}

void xputs (char *s)

{

while(*s)

{

putch (*s);

s++;

}

}

void xgets (char *s)

{

int i;

char ch ;

for (i = 0 ; i <= 98 ; i++)

{

ch = getche() ;

if (ch == '\r')


```
{
    *s = '\0';
    break;
}
if ( ch == '\b' )
{
    printf ( " \b" );
    i--;
    s--;
}
else
{
    *s = ch;
    s++;
}
}
}
```

- (b) Define a function **getint()**, which would receive a numeric string from the keyboard, convert it to an integer number and return the integer to the calling function. A sample usage of **getint()** is shown below:

```
#include <stdio.h>
int main( )
{
    int a;
    a = getint( );
    printf ( "you entered %d\n", a )
    return 0;
}
```

Program:

```
#include <stdio.h>
int getint( );
int main( )
```

```
{
    int a;

    printf ( "\nEnter a numeric string..." );
    a = getint( );
    printf ( "You entered %d\n", a );
    return 0;
}

int getint( )
{
    char str[ 6 ];
    int i, j, k, val;

    i = 0;
    while ( i <= 5 )
    {
        str[ i ] = getche( ); /* string input from keyboard */
        if ( str[ i ] == '\r' )
        {
            str[ i ] = '\0';
            break;
        }
        if ( str[ i ] == '\b' )
        {
            i--;
            printf ( " \b" );
        }
        else
            i++;
    }
    val = 0;
    k = 1;
    for ( j = i - 1; j >= 0; j-- )
    {
        val = val + ( str[ j ] - 48 ) * k; /* convert to numeric value */
        k = k * 10;
    }
}
```

```
return ( val );
```

2. Define a function `getfloat()`, which would receive a numeric string from the keyboard, convert it to a float value and return the float to the calling function.

Program:

* Program to convert a string to a float value and return the float to the calling function */

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

```
float getfloat();
```

```
int main()
```

```
{
```

```
    float floatresult = 0.0f;
```

```
    floatresult = getfloat();
```

```
    printf ( "After converting string to float, result is : " );
```

```
    printf ( "%0.2f\n", floatresult );
```

```
    return 0;
```

```
}
```

```
float getfloat()
```

```
{
```

```
    char str[] = "1234.56";
```

```
    float temp = 0.0f, result = 0.0f;
```

```
    int i, j, decimal;
```

```
    for ( i = 0; str[i] != '.'; i++ )
```

```
        result = ( result * 10 ) + str[i] - '0';
```

```
    decimal = 1;
```

```
    for ( j = i + 1; str[j] != '\0'; j++ )
```



```

    {
        temp = str[j] - '0';
        decimal = decimal * 10;
        temp = temp / decimal;
        result = result + temp;
    }
    return result;
}

```

- (d) If we are to display the following output properly aligned which format specifiers would you use?

Discovery of India	Jawaharlal Nehru	425.50
My Experiments with Truth	Mahatma Gandhi	375.50
Sunny Days	Sunil Gavaskar	95.50
One More Over	Erapalli Prasanna	85.00

Program:

```

#include <stdio.h>
int main()
{
    printf ( "%-30s%-20s%-10.2fn", "Discovery of India", "Jawaharlal
    Nehru", 425.50 );

    printf ( "%-30s%-20s%-10.2fn", "My Experiments with Truth",
    "Mahatma Gandhi", 375.50 );

    printf ( "%-30s%-20s%-10.2fn", "Sunny Days", "Sunil Gavaskar",
    95.50 );

    printf ( "%-30s%-20s%-10.2fn", "One More Over", "Erapalli
    Prasanna", 85.00 );

    return 0;
}

```