# CHAPTER FIFTEEN

# Strings

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

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

*Output:*

```
A
A
```

(b)
```
# include <stdio.h>
int main( )
{
    char  s[  ] = "Get organised! learn C!!" ;
    printf ( "%s\n", &s[ 2 ] ) ;
    printf ( "%s\n", s ) ;
    printf ( "%s\n", &s ) ;
    printf ( "%c\n", s[ 2 ] ) ;
    return 0 ;
```

```
}
```

*Output:*

```
t orgainsed ! learn c !!
get organised ! learn c !!
get orgainsed ! learn c !!
t
```

(c)
```
# include <stdio.h>
int main( )
{
    char  s[  ] = "No two viruses work similarly" ;
    int  i = 0 ;
    while ( s[ i ] != 0 )
    {
        printf ( "%c %c\n", s[ i ], *( s + i ) ) ;
        printf ( "%c %c\n", i[ s ], *( i + s ) ) ;
        i++ ;
    }
    return 0 ;
}
```

*Output:*

```
N  N
N  N
o  o
o  o
t   t
t   t
.....
.....
y   y
y   y
```

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

```
{
    char  s[ ] = "Churchgate: no church no gate" ;
    char  t[25] ;
    char  *ss, *tt ;
    ss = s ;
    while ( *ss != '\0' )
        *tt++ = *ss++ ;
    printf ( "%s\n", t ) ;
    return 0 ;
}
```

*Output:*

The code causes an exception as The variable 'tt' is being used without being initialized.

(e)
```
# include <stdio.h>
int main( )
{
    char str1[ ] = { 'H', 'e', 'l', 'l', 'o', 0 } ;
    char str2[ ] = "Hello" ;
    printf ( "%s\n", str1 ) ;
    printf ( "%s\n", str2 ) ;
    return 0 ;
}
```

*Output:*

```
Hello
Hello
```

(f)
```
# include <stdio.h>
void main( )
{.
    printf ( 5 + "Good Morning " ) ;
    return 0 ;
}
```

*Output:*

Morning

(g)
```
# include <stdio.h>
int main( )
{
    printf ( "%c\n", "abcdefgh"[ 4 ] ) ;
    return 0 ;
}
```

*Output:*

e

(h)
```
# include <stdio.h>
int main( )
{
    printf ( "%d %d %d\n", sizeof ( '3' ), sizeof ( "3" ), sizeof ( 3 ) ) ;
    return 0 ;
}
```

*Output:*

2 2 2

[B] Point out the errors, if any, in the following programs:

(a)
```
# include <stdio.h>
# include <string.h>
int main( )
{
    char *str1 = "United" ;
    char *str2 = "Front" ;
    char *str3 ;
    str3 = strcat ( str1, str2 ) ;
    printf ( "%s\n", str3 ) ;
    return 0 ;
```

```
}
```

No Error

(b)
```
# include <stdio.h>
int main( )
{
   int  arr[  ] = { 'A', 'B', 'C', 'D' } ;
   int  i ;
   for ( i = 0 ; i <= 3 ; i++ )
        printf ( "%d", arr[ i ] ) ;
   printf ( "\n" ) ;
   return 0 ;
}
```

No Error

(c)
```
# include <stdio.h>
int main( ) .
{
   char  arr[ 8 ] = "Rhombus" ;
   int  i ;
   for ( i = 0 ; i <= 7 ; i++ )
        printf ( "%d", *arr ) ;
        arr++ ;
   return 0 ;
}
```

Error. arr cannot be incremented.

[C] Fill in the blanks:

(a) "A" is a string whereas 'A' is a character.

(b) A string is terminated by a null character, which is written as \0.

(c) The array **char name[10]** can consist of a maximum of 9 characters.

(d) The array elements are always stored in consecutive memory locations.

**[D]** Attempt the following:

(a) Which is more appropriate for reading in a multi-word string?

gets( ) printf( ) scanf( ) puts( )

*Answer:*

gets( )

(b) If the string "Alice in wonder land" is fed to the following **scanf( )** statement, what will be the contents of the arrays **str1**, **str2**, **str3** and **str4**?

```
scanf ( "%s%s%s%s", str1, str2, str3, str4 ) ;
```

*Answer:*

str1 – Alice
str2 – in
str3 – wonder
str4 – land

(c) Write a program that extracts part of the given string from the specified position. For example, if the sting is "Working with strings is fun", then if from position 4, 4 characters are to be extracted then the program should return string as "king". If the number of characters to be extracted is 0 then the program should extract entire string from the specified position.

*Program:*

```
/* To extract a substring from a string */
# include <stdio.h>
# include <string.h>

int main( )
{
    char str[ 20 ], news[ 20 ] ;
    char *s, *t ;
    int pos, n, i ;

    printf ( "\nEnter the string: " ) ;
    scanf ( "%s", str ) ;

    printf ( "\nEnter the position and number of characters to extract: ",
              news ) ;
    scanf ( "%d %d", &pos, &n ) ;

    s = str ;
    t = news ;

    if ( n == 0 )
        n = strlen ( str ) ;

    s = s + pos - 1 ;

    for ( i = 0 ; i < n ; i++ )
    {
        *t = *s ;
        s++ ;
        t++ ;
    }
    *t = '\0' ;

    printf ( "\nThe substring is: %s\n", news ) ;

    return 0 ;
```

```
}
```

(d) Write a program that converts a string like "124" to an integer 124.

*Program:*

```
/* To convert a string to an integer */
/* A function atoi( ) converts string to an int */
# include <stdio.h>

int main( )
{
    char str[ 6 ] ;
    int num = 0, i ;

    printf ( "Enter a string containing a number: " ) ;
    scanf ( "%s", str ) ;

    for ( i = 0 ; str [ i ] != '\0' ; i++ )
    {
        if ( str[ i ] >= 48 && str[ i ] <= 57 )
            num = num * 10 + ( str[ i ] - 48 ) ;
        else
        {
            printf ( "Not a valid string\n" ) ;
            return 1 ;
        }
    }

    printf ( "\nThe number is: %d\n", num ) ;

    return 0 ;
}
```

(e) Write a program that generates and prints the Fibonacci words of order 0 through 5. If f(0) = "a", f(1) = "b", f(2) = "ba", f(3) = "bab", f(4) = "babba", etc.

*Program:*

```
/* Generate Fibonacci words of order 0 through 5 */
#include <stdio.h>
#include <string.h>

int main( )
{
   char str[ 50 ] ;
   char lastbutoneterm[ 50 ] = "A" ;
   char lastterm[ 50 ] = "B" ;
   int i ;

   for ( i = 1 ; i <= 5 ; i++ )
   {
       strcpy ( str, lastterm ) ;
       strcat ( str, lastbutoneterm ) ;
       printf ( "%s\n", str ) ;
       strcpy ( lastbutoneterm, lastterm );
       strcpy ( lastterm, str ) ;
   }

   return 0 ;
}
```

(f) To uniquely identify a book a 10-digit ISBN number is used. The rightmost digit is a checksum digit. This digit is determined from the other 9 digits using the condition that $d_1$ + $2d_2$ + $3d_3$ + ... + $10d_{10}$ must be a multiple of 11 (where $d_i$ denotes the $i^{th}$ digit from the right). The checksum digit $d_1$ can be any value from 0 to 10: the ISBN convention is to use the value X to denote 10. Write a program that receives a 10-digit

integer, computes the checksum, and reports whether the ISBN number is correct or not.

*Program:*

```
/* Check correctness of ISBN number */
#include <stdio.h>
#include <string.h>

int main( )
{
    char str[ 11 ] ;
    int i, j, sum ;

    printf ( "Enter 10 digit ISBN number: " ) ;
    scanf ( "%s", str ) ;

    j = 2 ;
    sum = 0 ;
    for ( i = 8 ; i >= 0 ; i-- )
    {
        sum = sum + ( str [ i ] - '0' ) * j ;
        j++ ;
    }

    for ( i = 0 ; i <= 9 ; i++ )
    {
        if ( ( sum + i ) % 11 == 0 )
            break ;
    }

    if ( i == str[ 9 ] - '0' )
        printf ( "ISBN Number is verified and found to be correct\n" ) ;
    else
        printf ( "Checksum error in ISBN Number\n" ) ;

    return 0 ;
```

}

(g) A Credit Card number is usually a 16-digit number. A valid Credit Card number would satisfy a rule explained below with the help of a dummy Credit Card number—4567 1234 5678 9129. Start with the rightmost - 1 digit and multiply every other digit by 2.

4567 1234 5678 9129

8 12 2 6 10 14 18 4

Then subtract 9 from any number larger than 10. Thus we get:

8 3 2 6 1 5 9 4

Add them all up to get 38.

Add all the other digits to get 42.

Sum of 38 and 42 is 80. Since 80 is divisible by 10, the Credit Card number is valid.

Write a program that receives a Credit Card number and checks using the above rule whether the Credit Card number is valid.

*Program:*

```
/* Verify correctness of Credit Card Number */
#include <stdio.h>
#include <string.h>

int main( )
{
    int len, i, sum, digit, multiple ;
    char str[ 20 ] ;
```

```
    printf ( "Enter the Credit Card number: " ) ;
    scanf ( "%s", str ) ;
    len = strlen ( str ) ;
    sum = 0 ;

    for ( i = 15 ; i >= 0 ; i-- )
    {
        digit = str[ i ] - '0';
        if ( i % 2 == 0 )
        {
            multiple = digit * 2 ;
            digit = multiple < 10 ? multiple : multiple - 9 ;
        }
        sum += digit ;
    }

    printf ( "%d\n", sum ) ;

    if ( sum % 10 == 0 )
        printf ( "Valid credit card number\n" ) ;
    else
        printf ( "Invalid credit card number\n" ) ;

    return 0 ;
}
```