Strings

Question 9.1

For the following statements will *arr[3]* and *ptr[3]* fetch the same character? [Yes/No]

```
char arr[ ] = "Surprised" ;
char *ptr = "Surprised" ;
```

Answer

Yes

Question 9.2

For the statements in 9.1 does the compiler fetch the character *arr[3]* and *ptr[3]* in the same manner?

Answer

No. For *arr[3]* the compiler generates code to start at location *arr*, move three locations past it, and fetch the character there. When it sees the expression *ptr[3]* it generates the code to start at location stored in *ptr*, add three to the pointer, and finally fetch the character pointed to.

In other words, *arr[3]* is three places past the start of the object named *arr*, whereas *ptr[3]* is three places past the object pointed to by *ptr*.

Question 9.3

Is there any difference in the following two statements?

```
char *ch = "Nagpur" ;
char ch[ ] = "Nagpur" ;
```


Answer

Yes. In the first statement, the character pointer *ch* stores the address of the string "Nagpur". The pointer *ch* can be made to point to some other character string (or even nowhere). The second statement, on the other hand, specifies that space for 7 characters be allocated and that the name of the location is *ch*. Thus, it specifies the size as well as initial values of the characters in array *ch*.

Question 9.4

When are *char a[ ]* and *char *a* treated as same by the compiler?

Answer

When using them as formal parameters while defining a function.

Question 9.5

Will the following program compile successfully?

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

Answer

No, because we can assign a new string to a pointer but not to an array.

Question 9.6

Which of the following is the correct output for the program given below?

```
#include <stdio.h>
int main( )
{
    int i ;
    char a[ ] = "\0" ;
    if ( printf ( "%s", a ) )
        printf ( "The string is not empty\n" ) ;
    else
        printf ( "The string is empty\n" ) ;
    return 0 ;
}
```

A. The string is not empty
B. The string is empty
C. No output
D. 0

Answer

B

Question 9.7

Which of the following is the correct output for the program given below?


```
#include <stdio.h>
int main( )
{
    int i ;
    char a[ ] = "\0" ;
    if ( printf ( "%s", a ) )
        printf ( "The string is empty\n" ) ;
    else
        printf ( "The string is not empty\n" ) ;
    return 0 ;
}
```

A. The string is empty
B. The string is not empty
C. No output
D. 0

Answer

B

Question 9.8

Which of the following is the correct output for the program given below?

```
#include <stdio.h>
int main( )
{
    char p[ ] = "%d\n" ;
    p[1] = 'c' ;
    printf ( p, 65 ) ;
    return 0 ;
}
```

A. A

B. a
C. c
D. 65

Answer

A

Question 9.9

Which of the following is the correct output for the program given below?

```
#include <stdio.h>
int main( )
{
    printf ( 5 + "Fascimile\n" ) ;
    return 0 ;
}
```

A. Error
B. Fascimile
C. mile
D. None of the above

Answer

C

Question 9.10

Which of the following is the correct output for the program given below?

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



```
void swap ( char *, char * ) ;
int main( )
{
    char *pstr[2] = {
                        "Hello",
                        "Good Morning"
                    } ;
    swap ( pstr[0], pstr[1] ) ;
    printf ( "%s\n%s", pstr[0], pstr[1] ) ;
    return 0 ;
}
void swap ( char *t1, char *t2 )
{
    char *t ;
    t = t1 ;
    t1 = t2 ;
    t2 = t ;
}
```

A. Good Morning
   Hello
B. Addresses of "Hello" and "Good Morning"
C. Hello
   Good Morning
D. Gello
   Hood Morning

Answer

C

Question 9.11

Which of the following is the correct output for the program given below?

```
#include <stdio.h>
#include <string.h>
int main( )
{
    char str1[5], str2[5] ;
    int i ;
    gets ( str1 ) ;
    gets ( str2 ) ;
    i = strcmp ( str1, str2 ) ;
    printf ( "%d\n", i ) ;
    return 0 ;
}
```

A. Unpredictable integer value
B. 0
C. –1
D. Error

Answer

A

Question *9.12*

Which of the following is the correct output for the program given below?

```
#include <stdio.h>
int main( )
{
    char str1[ ] = "Hello" ;
    char str2[ ] = "Hello" ;
    if ( str1 == str2 )
        printf ( "Equal\n" ) ;
    else
        printf ( "Unequal\n" ) ;
    return 0 ;
}
```

A. Equal
B. Unequal
C. Error
D. None of the above

Answer

B

Question *9.13*

Which of the following is the correct output for the program given below?

```
#include <stdio.h>
int main( )
{
    char t ;
    char *p1 = "Harder you work", *p2 ;
    p2 = p1 ;
    p1 = "Luckier you get" ;
    printf ( "%s %s\n", p1, p2 ) ;
    return 0 ;
}
```

A. Harder you work Luckier you get
B. Luckier you get Harder you work
C. Harder you work Harder you work
D. Luckier you get Luckier you get

Answer

B

Question 9.14

What will be the output of the following program?

```
#include <stdio.h>
int main( )
{
    printf ( "%u %s\n", &"Hello", &"Hello" ) ;
    return 0 ;
}
```

A. 1760 Hello
B. Hello 1760
C. Hello Hello
D. Error

Answer

A

Question 9.15

Which of the following is the correct output for the program given below?

```
#include <stdio.h>
int main( )
{
    char str[10] = "Angel" ;
    str[6] = 'd' ;
    printf ( "%s\n", str ) ;
    return 0 ;
}
```

A. Angel d
B. d

C. Angel
D. Error

Answer

C

Question 9.16

Which of the following is the correct output for the program given below?

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

A. Error
B. Come September
C. Base address of *str*
D. No output

Answer

A

Question 9.17

Which of the following is the correct output for the program given below?

```
#include <stdio.h>
int main( )
{
    printf ( "Icecream " "Chocolate pie\n" ) ;
    return 0 ;
}
```

A. Error
B. Icecream Chocolate pie
C. Icecream
D. Chocoloate pie

Answer

B

Question *9.18*

Which of the following is the correct output for the program given below?

```
#include <stdio.h>
int main( )
{
    char str[25] = "Catch me, if you can!" ;
    printf ( "%s\n", &str + 2 ) ;
    return 0 ;
}
```

A. Garbage value
B. Error
C. No output
D. tch me, if you can!

Answer

A

Question *9.19*

What will be the output of the following program if characters 'a, 'b' and 'c' and enter are supplied to it?

```
#include <stdio.h>
int main( )
{
    void fun( ) ;
    fun( ) ;
    printf ( "\n" ) ;
    return 0 ;
}
void fun( )
{
    char c ;
    if ( ( c = getchar( ) ) != '\n' )
        fun( ) ;
    printf ( "%c", c ) ;
}
```

A. abc abc
B. bca
C. Infinite loop
D. cba

Answer

D

Question 9.20

Which of the following is the correct output for the program given below?

```
#include <stdio.h>
# include <string.h>
int main( )
{
    char str1[ ] = "Hello" ;
    char str2[10] ;
    char *t, *s ;
    s = str1 ;
    t = str2 ;
    while ( *t = *s )
        *t++ = *s++ ;
    printf ( " %s\n", str2 ) ;
    return 0 ;
}
```

A. Hello
B. HelloHello
C. No output
D. Ello

Answer

A

Question 9.21

Which of the following is the correct output for the program given below?

```
#include <stdio.h>
#include <string.h>
int main( )
{
    char str1[20] = "Hello", str2[20] = " World" ;
    printf ( "%s\n", strcpy ( str2, strcat ( str1, str2 ) ) ) ;
    return 0 ;
}
```

A. Hello
B. World
C. Hello World
D. WorldHello

Answer

C

Question 9.22

Which of the following is the correct output for the program given below?

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

A. man
B. sales
C. sales man
D. sales\0man

Answer

B

Question 9.23

Which of the following is the correct output for the program given below?

```
#include <stdio.h>
int main( )
{
    char str[ ] = "sales\0man\0" ;
    printf ( "%d\n", sizeof ( str ) ) ;
    return 0 ;
}
```

A. 10
B. 6
C. 5
D. 11

Answer

D

Question 9.24

Which of the following is the correct output for the program given below?

```
#include <stdio.h>
#include <string.h>
int main( )
{
    char str[ ] = "sales\0\man\0" ;
```



```
    printf ( "%d\n", strlen ( str ) ) ;
    return 0 ;
}
```

A. 10
B. 6
C. 5
D. 11

Answer

C

Question 9.25

If size of an integer is 4 bytes, which of the following is the correct output for the program given below?

```
#include <stdio.h>
#include <string.h>
int main( )
{
    printf ( "%d\n", strlen ( "123456" ) ) ;
    return 0 ;
}
```

A. 6
B. 12
C. 7
D. 2

Answer

A

Question 9.26

Which of the following is the correct output for the program given below?

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

A. Error
B. d
C. e
D. abcdefgh

Answer

C

Question 9.27

Which of the following is the correct output for the program given below?

```
#include <stdio.h>
int main( )
{
    char str[7] = "Strings" ;
    printf ( "%s\n", str ) ;
    return 0 ;
}
```

A. Error. Array bounds overflow.
B. Strings

C. Cannot predict
D. None of the above

Answer

A. C. Here *str[ ]* has been declared as a 7 character array and into it a 8 character string has been stored. This will result into overwriting of the byte beyond the seventh byte reserved for the array with a '\0'. There is always a possibility that something important gets overwritten which will be unsafe.

Question 9.28

How will you output \n on the screen?

Answer

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

Question 9.29

If sizes of a *char*, an *int* and a *float* are 1, 4 and 4 bytes respectively, which of the following is the correct output for the program given below?

```
#include <stdio.h>
int main( )
{
    char ch = 'A' ;
    printf ( "%d %d %d", sizeof ( ch ), sizeof  ( sizeof ( 'A' ) ), sizeof ( 3.14 )) ;
    printf ( "\n" ) ;
    return 0 ;
}
```

A. 1 2 4

B. 1 4 8
C. 2 2 4
D. 2 4 8

Answer

B

Question 9.30

Which of the following is the correct output for the program given below?

```
#include <stdio.h>
int main( )
{
    printf ( "%d %d %d\n", sizeof ( 3.0f ), sizeof ( "3" ), sizeof ( 3.0 ) ) ;
    return 0 ;
}
```

A. 8 1 4
B. 4 2 8
C. 4 2 4
D. 10 3 4

Answer

B

Question 9.31

Is the following program correct? [Yes/No]

```
#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 ;
}
```


Answer

No, since what is present in memory beyond 'United' is not known and we are attaching 'Front' at the end of 'United', thereby overwriting something, which is an unsafe thing to do.

Question 9.32

How will you improve the code in 9.31 above?

Answer

```
#include <stdio.h>
#include <string.h>
int main( )
{
    char str1[15] = "United" ;
    char *str2 = "Front" ;
    char *str3 ;
    str3 = strcat ( str1, str2 ) ;
    printf ( "%s\n", str3 ) ;
    return 0 ;
}
```

Question 9.33

In the following code which function will get called—the user-defined *strcpy( )* or the one in the standard library?

```
#include <stdio.h>
#include <string.h>
void strcpy ( char *,char * ) ;
int main( )
{
    char str1[ ] = "Keep India Beautiful... Immigrate!" ;
    char str2[40] ;
    strcpy ( str2, str1 ) ;
    printf ( "%s\n", str2 ) ;
    return 0 ;
}
void strcpy ( char *t, char *s )
{
    while ( *s )
    {
        *t = *s ;
        t++ ;
        s++ ;
    }
    *t = '\0' ;
}
```

Answer

User-defined *strcpy( )*.

Question *9.34*

Can you compact the code in *strcpy( )* function given below into one line?

```
void strcpy ( char *t, char *s )
{
    while ( *s )
    {
        *t = *s ;
        t++ ;
        s++ ;
    }
    *t = '\0' ;
}
```

Answer

```
void strcpy ( char *t, char *s )
{
    while ( *t++ = *s++ ) ;
}
```

Question *9.35*

If size of a pointer is 4 bytes then what will be the output of the following program?

```
#include <stdio.h>
#include <string.h>
int main( )
{
    char *str[ ] = { "Frogs", "Do", "Not", "Die.", "They", "Croak!" } ;
    printf ( "%d %d\n", sizeof ( str ), strlen ( str[0] ) ) ;
    return 0 ;
}
```

Answer

24 5

Question *9.36*

How will you find the length of each string in the program 9.35 above?

Answer

```
#include <stdio.h>
#include <string.h>
int main( )
{
    char *str[ ] = { "Frogs", "Do", "Not", "Die.", "They", "Croak!" } ;
    int i ;
    for ( i = 0 ; i <= 5 ; i++ )
        printf ( "%s %d\n", str[ i ], strlen ( str[ i ] ) ) ;
    return 0 ;
}
```

Question 9.37

What is the difference in the following declarations?

```
char *p = "Samuel" ;
char a[ ] = "Samuel" ;
```

Answer

Here *a* is an array big enough to hold the message and the '\0' following the message. Individual characters within the array can be changed but the address of the array will remain same.

On the other hand, *p* is a pointer, initialized to point to a string constant. The pointer *p* may be modified to point to another string, but if you attempt to modify the string at which *p* is pointing the result is undefined.

Question 9.38

While handling a string do we always have to process it character by character or there exists a method to process the entire string as one unit.

Answer

A string can be processed only on a character-by-character basis.

Question 9.39

Which of the following function is correct that finds the length of a string?

A.
```
int xstrlen ( char *s )
{
    int length = 0 ;
    while ( *s != '\0' )
    {
        length++ ; s++ ;
    }
    return ( length ) ;
}
```

B.
```
int xstrlen ( char s )
{
    int length = 0 ;
    while ( s != '\0' )
    {
        length++ ; s++ ;
    }
    return ( length ) ;
}
```

C.
```
int xstrlen ( char *s )
{
    int length = 0 ;
    while ( *s != '\0' )
    {
        length++ ;
    }
```

```
    return ( length ) ;
}
```

D.
```
int xstrlen ( char *s )
{
    int length = 0 ;
    while ( *s != '\0' )
    {
        s++ ;
    }
    return ( length ) ;
}
```

Answer

A

Question *9.40*

Which of the following is the correct output for the program given below?

```
#include <stdio.h>
#include <string.h>
int main( )
{
    static char s[ ] = "Hello !" ;
    printf ( "%d\n", *( s + strlen ( s ) ) ) ;
    return 0 ;
}
```

A. 8
B. 0
C. 16
D. Error

Answer

B

Question *9.41*

Which of the following function can be used to find the first occurrence of a given string in another string?

A. strchr( )
B. strrchr( )
C. strstr( )
D. strnset( )

Answer

C

Question *9.42*

Which of the following is the correct output for the program given below?

```
#include <stdio.h>
#include <string.h>
int main( )
{
    char sentence[ 80 ] ;
    int i ;
    printf ( "Enter a line of text\n" ) ;
    gets ( sentence ) ;
    for ( i = strlen ( sentence ) - 1 ; i >= 0 ; i-- )
        putchar ( sentence[ i ] ) ;
    return 0 ;
}
```

A. The sentence will get printed in the same order as it is entered.
B. The sentence will get printed in reverse order.
C. Half of the sentence will get printed.
D. None of above.

Answer

B

Question 9.43

The library function used to reverse a string is

A. strstr( )
B. strrev( )
C. revstr( )
D. None of the above

Answer

B

Question 9.44

Which of the following is the correct output for the program given below?

```
#include <stdio.h>
#include <string.h>
int main( )
{
    static char str1[ ] = "dills" ;
    static char str2[20] ;
    static char str3[20] = "Daffo" ;
    int l ;
    l = strcmp ( strcat ( str3, strcpy ( str2, str1 ) ), "Daffodills" ) ;
```



```
    printf ( "%d\n", l ) ;
    return 0 ;
}
```

A. 0
B. 1
C. 2
D. Cannot be determined

Answer

A

Question 9.45

Which of the following function is more appropriate for reading in a multi-word string?

A. printf( )
B. scanf( )
C. gets( )
D. puts( )

Answer

C

Question 9.46

Which of the following is the correct output for the program given below?

```
#include <stdio.h>
int main( )
{
    static char mess[6][30] = {
```

```
                                "Don't Walk in front of me...",
                                "I may not follow;",
                                "Don't walk behind me...",
                                "Just walk beside me...",
                                "And be my friend."
                               };
    printf ( "%c %c\n", *( mess[2] + 9 ), * ( * ( mess + 2 ) + 9 ) ) ;
    return 0 ;
}
```

A. t t
B. k k
C. n k
D. m f

Answer

B

Question *9.47*

Which of the following function sets first *n* characters of a string to a given character?

A. strinit( )
B. strnset( )
C. strset( )
D. strcset( )

Answer

B

Question *9.48*

Which of the following is the correct output for the program given below?


```
#include <stdio.h>
int main( )
{
    char str[ ]="Nagpur";
    str [0] = 'K';
    printf ( "%s", str ) ;
    str = "Kanpur" ;
    printf ( "%s\n", str + 1 ) ;
    return 0 ;
}
```

A. Kagpur Kanpur
B. Nagpur Kanpur
C. Kagpur anpur
D. Error

Answer

D

Question *9.49*

The library function used to find the last occurrence of a character in a string is:

A. strnstr( )
B. laststr( )
C. strchr( )
D. strstr( )

Answer

C

Question 9.50

Which of the following is the correct output for the program given below?

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

A. Good Morning
B. Good
C. M
D. Morning

Answer

D

Question 9.51

If the given two strings are identical, then *strcmp( )* function returns

A. -1
B. 1
C. 0
D. Address of the first string

Answer

C

Question 9.52

Which of the following statements are correct about the following program?

```
#include <stdio.h>
int main( )
{
    char str[20], *s ;
    printf ( "Enter a string\n" ) ;
    scanf ( "%s", str ) ;
    s = str ;
    while ( *s != '\0' )
    {
        if ( *s >= 97 && *s <= 122 )
            *s = *s - 32 ;
        s++ ;
    }
    return 0 ;
}
```

A. The code converts a string to an integer.
B. The code converts lower case character to upper case.
C. The code converts upper case characters to lower case.
D. None of the above.

Answer

B

Question 9.53

Which of the following statements are correct?

A. A string is a collection of characters terminated by '\0'.
B. The format specifier *%s* is used to print a string.

C. The length of a string can be obtained using *strlen( )*.
D. The pointers CANNOT work on string.

Answer

A, B, C

Question 9.54

Which of the following is the correct output for the program given below?

```
#include <stdio.h>
int main( )
{
    char *names[ ]= {
                        "Roshni",
                        "Manish",
                        "Sona",
                        "Baiju",
                        "Ritu"
                    };
    int i ;
    char *t ;
    t = names[3] ;
    names[3] = names[4] ;
    names[4] = t ;
    for ( i = 0 ; i <= 4 ; i++ )
        printf ( "%s", names[ i ] ) ;
    printf ( "\n" ) ;
    return 0 ;
}
```

A. Roshni Manish Sona Baiju Ritu
B. Roshni Manish Sona Ritu Baiju
C. Roshni Manish Baiju Sona Ritu

D. Roshni Manish Ritu Sona Baiju

Answer

B

Question 9.55

Which of the following statements is correct?

A. *strcmp (s1 , s2)* returns a number less then 0 if $s1 > s2$
B. *strcmp (s1 , s2)* returns a number greater then 0 if $s1 < s2$
C. *strcmp (s1 , s2)* returns 0 if $s1 == s2$
D. *strcmp (s1 , s2)* returns 1 if $s1 == s2$

Answer

C

Question 9.56

Which of the following is the correct output for the program given below?

```
#include <stdio.h>
int main( )
{
    static char s[25] = "The cocaine man";
    int i = 0 ;
    char ch ;
    ch = s[++i] ;
    printf ( "%c", ch ) ;
    ch = s[i++] ;
    printf ( "%c", ch ) ;
    ch = i++[s] ;
    printf ( "%c", ch ) ;
```

```
    ch = ++i[s] ;
    printf ( "%c\n", ch ) ;
    return 0 ;
}
```

A. hhe!
B. he c
C. The c
D. Hhec

Answer

A

Question 9.57

Which of the following statements are correct about the C declarations given below?

```
char *p = "Sanjay" ;
char a[ ] = "Sanjay" ;
```

A. There is no difference in the declarations and both serve the same purpose.
B. The first statement is incorrect since '\0' is not given in the string, whereas the second is incorrect as size of array is not mentioned.
C. *p* is a non-const pointer pointing to a non-const string, whereas *a* is a const pointer pointing to a non-const string.
D. The pointer *p* can be modified to point to another string, whereas individual characters within array *a* can be changed.
E. In both cases the '\0' will be added at the end of the string "Sanjay".

Answer

C, D, E

Question 9.58

If size of a pointer is 32 bits what will be the output of the following program?

```
#include <stdio.h>
int main( )
{
    char a[ ] = "Visual C++" ;
    char *b = "Visual C++" ;
    printf ( "%d %d\n", sizeof ( a ), sizeof ( b ) ) ;
    printf ( "%d %d\n", sizeof ( *a ), sizeof ( *b ) ) ;
    return 0 ;
}
```

Answer

11 4
1 1

Question 9.59

How will you pass the number elements present in an array to a function in a generic way?

Answer

We should use the *sizeof* operator to obtain the number of elements as shown below:

```
Int arr[ ] = { 10, 20, 30, 40, 50 } ;
fun ( sizeof ( arr ) / sizeof ( arr[ 0 ] ) ) ;
```

Question *9.60*

Which of the following expressions would yield the value 100 for an array defined below?

```
Int arr[ ][ 2 ][ 3 ] = {
                        {
                            { 100, 200, 300 },
                            { 400, 500, 600 }
                        },
                        {
                            { 10, 20, 30 },
                            { 40, 50, 60 }
                        }
                        {
                            { 1, 2, 3 },
                            { 4, 5, 6 }
                        }
                    };
```

A. arr[0][0][0]
B. **arr[0]
C. ***arr
D. *arr[0][0]

Answer

A, B, C, D