1. 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: Option **B**
Explanation:
Declaration:
`char *strnset(char *s, int ch, size_t n);` Sets the first `n` characters of `s` to `ch`

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

int main(void)
{
   char *string = "abcdefghijklmnopqrstuvwxyz";
   char letter = 'x';

   printf("string before strnset: %s\n", string);
   strnset(string, letter, 13);
   printf("string after  strnset: %s\n", string);

   return 0;
}
```

Output:

string before strnset: abcdefghijklmnopqrstuvwxyz

string after strnset: xxxxxxxxxxxxxnopqrstuvwxyz

View Answer Discuss in Forum Workspace Report

2. If the two strings are identical, then `strcmp()` function returns

   A. -1

   B. 1

   C. 0

   D. Yes

Answer: Option **C**
Explanation:
Declaration: `strcmp(const char *s1, const char*s2);`
The `strcmp` return an `int` value that is

if s1 < s2 returns a value < 0

if s1 == s2 returns 0

if s1 > s2 returns a value > 0

View Answer Discuss in Forum Workspace Report

3. How will you print \n on the screen?

   A. printf("\n");

   B. echo "\\n";

   C. printf('\n');

   D. printf("\\n");

Answer: Option **D**
Explanation:
The statement `printf("\\n");` prints '\n' on the screen.
View Answer Discuss in Forum Workspace Report

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

   A. strnstr()

   B. laststr()

   C. strrchr()

   D. strstr()

Answer: Option **C**
Explanation:
Declaration: `char *strrchr(const char *s, int c);`
It scans a string `s` in the reverse direction, looking for a specific character `c`.
Example:

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

int main(void)
{
   char text[] = "I learn through IndiaBIX.com";
   char *ptr, c = 'i';

   ptr = strrchr(text, c);
   if (ptr)
      printf("The position of '%c' is: %d\n", c, ptr-text);
   else
      printf("The character was not found\n");
   return 0;
}
```

Output:

```
The position of 'i' is: 19
```

View Answer Discuss in Forum Workspace Report

5. Which of the following function is used to find the first occurrence of a given string in another string?

   A. strchr()

   B. strrchr()

   C. strstr()

   D. strnset()

   Answer: Option **C**
   Explanation:
   The function `strstr()` Finds the first occurrence of a substring in another string
   Declaration: char *strstr(const char *s1, const char *s2);
   Return Value:
   On success, `strstr` returns a pointer to the element in `s1` where `s2` begins (points to `s2` in `s1`).
   On error (if `s2` does not occur in `s1`), `strstr` returns null.
   Example:

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

int main(void)
{
   char *str1 = "IndiaBIX", *str2 = "ia", *ptr;

   ptr = strstr(str1, str2);
   printf("The substring is: %s\n", ptr);
   return 0;
}
```

   Output: The substring is: iaBIX

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

   A. printf();

   B. scanf();

   C. gets();

   D. puts();

   Answer: Option **C**
   Explanation:
   `gets();` collects a string of characters terminated by a new line from the standard input stream `stdin`

```
#include <stdio.h>

int main(void)
{
   char string[80];

   printf("Enter a string:");
   gets(string);
```

```
    printf("The string input was: %s\n", string);
    return 0;
}
```

Output:

Enter a string: IndiaBIX

The string input was: IndiaBIX

View Answer Discuss in Forum Workspace Report

7. 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: Option **A**

Explanation:

Option A is the correct function to find the length of given string.

Example:

```
#include<stdio.h>

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

int main()
{
    char d[] = "IndiaBIX";
    printf("Length = %d\n", xstrlen(d));
    return 0;
}
```

Output: Length = 8

1. What will be the output of the program ?

```
#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: Option **C**

Explanation:

Step 1: `char str1[20] = "Hello", str2[20] = " World";` The variable `str1` and `str2` is declared as an array of characters and initialized with value "Hello" and " World" respectively.

Step 2: `printf("%s\n", strcpy(str2, strcat(str1, str2)));`

=> `strcat(str1, str2))` it append the string `str2` to `str1`. The result will be stored in `str1`. Therefore `str1` contains "Hello World".

=> `strcpy(str2, "Hello World")` it copies the "Hello World" to the variable `str2`.

Hence it prints "Hello World".

View Answer Discuss in Forum Workspace Report

2. What will be the output of the program ?

```
#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: Option **A**

Explanation:

Step 1: `char p[] = "%d\n";` The variable `p` is declared as an array of characters and initialized with string "%d".

Step 2: `p[1] = 'c';` Here, we overwrite the second element of array `p` by 'c'. So array `p` becomes "%c".

Step 3: `printf(p, 65);` becomes `printf("%c", 65);`

Therefore it prints the ASCII value of 65. The output is 'A'.

View Answer Discuss in Forum Workspace Report

3. What will be the output of the program ?

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

int main()
{
    printf("%d\n", strlen("123456"));
    return 0;
}
```

A. 6

B. 12

C. 7

D. 2

Answer: Option **A**

Explanation:

The function `strlen` returns the number of characters in the given string.

Therefore, `strlen("123456")` returns 6.

Hence the output of the program is "6".

View Answer Discuss in Forum Workspace Report

4. What will be the output of the program ?

```
#include<stdio.h>

int main()
{
    printf(5+"Good Morning\n");
    return 0;
}
```

A. Good Morning

B. Good

C. M

D. Morning

Answer: Option **D**
Explanation:
`printf(5+"Good Morning\n");` It skips the 5 characters and prints the given string.

Hence the output is "Morning"

View Answer Discuss in Forum Workspace Report

5. What will be the output of the program ?

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

int main()
{
    char str[] = "India\0\BIX\0";
    printf("%s\n", str);
    return 0;
}
```

A. BIX

B. India

C. India BIX

D. India\0BIX

Answer: Option **B**
Explanation:

A string is a collection of characters terminated by '\0'.

Step 1: `char str[] = "India\0\BIX\0";` The variable str is declared as an array of characters and initialized with value "India"
Step 2: `printf("%s\n", str);` It prints the value of the `str`.

The output of the program is "India".

6. What will be the output of the program If characters 'a', 'b' and 'c' enter are supplied as input?

```
#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: Option **D**
Explanation:
Step 1: `void fun();` This is the prototype for the function `fun()`.
Step 2: `fun();` The function `fun()` is called here.
The function `fun()` gets a character input and the input is terminated by an enter key(New line character). It prints the given character in the reverse order.

The given input characters are "abc"

Output: cba
View Answer Discuss in Forum Workspace Report

7. What will be the output of the program ?

```
#include<stdio.h>

int main()
```

```
{
    printf("India", "BIX\n");
    return 0;
}
```

A. Error

B. India BIX

C. India

D. BIX

Answer: Option **C**

Explanation:

`printf("India", "BIX\n");` It prints "India". Because `,`(comma) operator has Left to Right associativity. After printing "India", the statement got terminated.

View Answer Discuss in Forum Workspace Report

8. What will be the output of the program ?

```
#include<stdio.h>

int main()
{
    char str[7] = "IndiaBIX";
    printf("%s\n", str);
    return 0;
}
```

A. Error

B. IndiaBIX

C. Cannot predict

D. None of above

Answer: Option **C**

Explanation:

Here `str[]` has declared as 7 character array and into a 8 character is stored. This will result in overwriting of the byte beyond 7 byte reserved for `'\0'`.

View Answer Discuss in Forum Workspace Report

9. What will be the output of the program ?

```
#include<stdio.h>

int main()
{
    char *names[] = { "Suresh", "Siva", "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]);
    return 0;
}
```

A. Suresh, Siva, Sona, Baiju, Ritu

B. Suresh, Siva, Sona, Ritu, Baiju

C. Suresh, Siva, Baiju, Sona, Ritu

D. Suresh, Siva, Ritu, Sona, Baiju

Answer: Option **B**
Explanation:
Step 1: `char *names[] = { "Suresh", "Siva", "Sona", "Baiju", "Ritu"};` The variable `names` is declared as an pointer to a array of strings.
Step 2: `int i;` The variable `i` is declared as an integer type.
Step 3: `char *t;` The variable `t` is declared as pointer to a string.
Step 4: `t = names[3]; names[3] = names[4]; names[4] = t;` These statements the swaps the 4 and 5 element of the array `names`.
Step 5: `for(i=0; i<=4; i++) printf("%s,", names[i]);` These statement prints the all the value of the array `names`.

Hence the output of the program is "Suresh, Siva, Sona, Ritu, Baiju".

View Answer Discuss in Forum Workspace Report

10. What will be the output of the program ?

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

int main()
{
    char str[] = "India\0\BIX\0";
    printf("%d\n", strlen(str));
    return 0;
}
```

A. 10

B. 6

C. 5

D. 11

Answer: Option **C**
Explanation:
The function `strlen` returns the number of characters int the given string.
Therefore, `strlen(str)` becomes `strlen("India")` contains 5 characters. A string is a collection of characters terminated by '\0'.

The output of the program is "5

11. What will be the output of the program ?

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

int main()
{
    static char str1[] = "dills";
    static char str2[20];
    static char str3[] = "Daffo";
    int i;
    i = strcmp(strcat(str3, strcpy(str2, str1)), "Daffodills");
    printf("%d\n", i);
    return 0;
}
```

A. 0

B. 1

C. 2

D. 4

Answer: Option **A**
Explanation:
No answer description available for this question. Let us discuss.
View Answer Discuss in Forum Workspace Report

12. What will be the output of the program ?

```
#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: Option **B**

Explanation:
No answer description available for this question. Let us discuss.
View Answer Discuss in Forum Workspace Report

13. What will be the output of the program ?

```
#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", ch);
    return 0;
}
```

A. hhe!

B. he c

C. The c

D. Hhec

Answer: Option **A**
Explanation:
No answer description available for this question. Let us discuss.
View Answer Discuss in Forum Workspace Report

14. What will be the output of the program in 16-bit platform (Turbo C under DOS) ?

```
#include<stdio.h>

int main()
{
    printf("%d, %d, %d", 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: Option **B**
Explanation:
Step 1:
`printf("%d, %d, %d", sizeof(3.0f), sizeof('3'), sizeof(3.0));`

The sizeof function returns the size of the given expression.

`sizeof(3.0f)` is a floating point constant. The size of `float` is 4 bytes
`sizeof('3')` It converts '3' in to ASCII value.. The size of `int` is 2 bytes
`sizeof(3.0)` is a double constant. The size of `double` is 8 bytes

Hence the output of the program is 4,2,8

Note: The above program may produce different output in other platform due to the platform dependency of C compiler.

In Turbo C, 4 2 8. But in GCC, the output will be 4 4 8.

View Answer Discuss in Forum Workspace Report

15. What will be the output of the program ?

```
#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: Option **B**
Explanation:

The function printf() returns the number of charecters printed on the console.

Step 1: `char a[] = "\0";` The variable `a` is declared as an array of characters and it initialized with "\0". It denotes that the string is empty.
Step 2: `if(printf("%s", a))` The `printf()` statement does not print anything, so it returns '0'(zero). Hence the if condition is failed.
In the `else` part it prints "The string is not empty".

16.Char=1, int=4, and float=4 bytes size, What will be the output of the program ?

```
#include<stdio.h>

int main()
{
    char ch = 'A';
    printf("%d, %d, %d", sizeof(ch), sizeof('A'), sizeof(3.14f));
    return 0;
}
```

A. 1, 2, 4

B. 1, 4, 4

C. 2, 2, 4

D. 2, 4, 8

Answer: Option **B**

Explanation:

Step 1: `char ch = 'A';` The variable `ch` is declared as an character type and initialized with value 'A'.

Step 2:

`printf("%d, %d, %d", sizeof(ch), sizeof('A'), sizeof(3.14));`

The sizeof function returns the size of the given expression.

`sizeof(ch)` becomes `sizeof(char)`. The size of `char` is 1 byte.
`sizeof('A')` becomes `sizeof(65)`. The size of `int` is 4 bytes (as mentioned in the question).
`sizeof(3.14f)`. The size of `float` is 4 bytes.

Hence the output of the program is 1, 4, 4

View Answer Discuss in Forum Workspace Report

17. If the size of pointer is 32 bits What will be the output of the 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", sizeof(*a), sizeof(*b));
    return 0;
}
```

A. 10, 2
2, 2

B. 10, 4
1, 2

C. 11, 4
1, 1

D. 12, 2
2, 2

Answer: Option **C**

Explanation:

No answer description available for this question. Let us discuss.

View Answer Discuss in Forum Workspace Report

18. What will be the output of the program ?

```
#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: Option **B**

Explanation:

No answer description available for this question. Let us discuss.

View Answer Discuss in Forum Workspace Report

19. What will be the output of the program ?

```
#include<stdio.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: Option **A**
Explanation:
No answer description available for this question. Let us discuss.
View Answer Discuss in Forum Workspace Report

20. What will be the output of the program ?

```
#include<stdio.h>

int main()
{
    char str[] = "India\0BIX\0";
    printf("%d\n", sizeof(str));
    return 0;
}
```

A. 10

B. 6

C. 5

D. 11

Answer: Option **D**
Explanation:

The following examples may help you understand this problem:

1. `sizeof("")` returns 1 (1*).
2. `sizeof("India")` returns 6 (5 + 1*).
3. `sizeof("BIX")` returns 4 (3 + 1*).
4. `sizeof("India\0BIX")` returns 10 (5 + 1 + 3 + 1*).
   Here '\0' is considered as 1 char by sizeof() function.
5. `sizeof("India\0BIX\0")` returns 11 (5 + 1 + 3 + 1 + 1*).
   Here '\0' is considered as 1 char by sizeof() function.

21. What will be the output of the program ?

```
#include<stdio.h>

int main()
```

```
{
    char str[25] = "IndiaBIX";
    printf("%s\n", &str+2);
    return 0;
}
```

A. Garbage value

B. Error

C. No output

D. diaBIX

Answer: Option **A**

Explanation:

Step 1: `char str[25] = "IndiaBIX";` The variable `str` is declared as an array of characteres and initialized with a string "IndiaBIX".

Step 2: `printf("%s\n", &str+2);`

=> In the printf statement `%s` is string format specifier tells the compiler to print the string in the memory of `&str+2`

=> `&str` is a location of string "IndiaBIX". Therefore `&str+2` is another memory location.

Hence it prints the Garbage value.

View Answer Discuss in Forum Workspace Report

22. What will be the output of the program ?

```
#include<stdio.h>

int main()
{
    char str = "IndiaBIX";
    printf("%s\n", str);
    return 0;
}
```

A. Error

B. IndiaBIX

C. Base address of `str`

D. No output

Answer: Option **A**

Explanation:

The line `char str = "IndiaBIX";` generates "Non portable pointer conversion" error.

To eliminate the error, we have to change the above line to

`char *str = "IndiaBIX";` (or) `char str[] = "IndiaBIX";`

Then it prints "IndiaBIX".

View Answer Discuss in Forum Workspace Report

23. What will be the output of the program ?

```
#include<stdio.h>

int main()
{
    char str[] = "Nagpur";
    str[0]='K';
    printf("%s, ", str);
    str = "Kanpur";
    printf("%s", str+1);
    return 0;
}
```

A. Kagpur, Kanpur

B. Nagpur, Kanpur

C. Kagpur, anpur

D. Error

Answer: Option **D**

Explanation:

The statement `str = "Kanpur";` generates the LVALUE required error. We have to use strcpy function to copy a string.

To remove error we have to change this statement `str = "Kanpur";` to `strcpy(str, "Kanpur");`

The program prints the string "anpur"

View Answer Discuss in Forum Workspace Report

24. What will be the output of the program ?

```
#include<stdio.h>

int main()
{
    printf(5+"IndiaBIX\n");
    return 0;
}
```

A. Error

B. IndiaBIX

C. BIX

D. None of above

Answer: Option **C**

Explanation:

`printf(5+"IndiaBIX\n");` In the printf statement, it skips the first 5 characters and it prints "BIX"

View Answer Discuss in Forum Workspace Report

25. What will be the output of the program ?

```
#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 same order as it entered

B. The sentence will get printed in reverse order

C. Half of the sentence will get printed

D. None of above

Answer: Option **B**

Explanation:

No answer description available for this question. Let us discuss.

26. What will be the output of the program ?

```
#include<stdio.h>
void swap(char *, char *);

int main()
{
    char *pstr[2] = {"Hello", "IndiaBIX"};
    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. IndiaBIX
Hello

B. Address of "Hello" and "IndiaBIX"

C. Hello
IndiaBIX

D. Iello
HndiaBIX

Answer: Option **C**
Explanation:
Step 1: `void swap(char *, char *);` This prototype tells the compiler that the function swap accept two strings as arguments and it does not return anything.
Step 2: `char *pstr[2] = {"Hello", "IndiaBIX"};` The variable `pstr` is declared as an pointer to the array of strings. It is initialized to
`pstr[0] = "Hello"`, `pstr[1] = "IndiaBIX"`
Step 3: `swap(pstr[0], pstr[1]);` The `swap` function is called by "call by value". Hence it does not affect the output of the program.
If the `swap` function is "called by reference" it will affect the variable `pstr`.
Step 4: `printf("%s\n%s", pstr[0], pstr[1]);` It prints the value of `pstr[0]` and `pstr[1]`.

Hence the output of the program is

Hello
IndiaBIX
View Answer Discuss in Forum Workspace Report

---

27. What will be the output of the program (Turbo C in 16 bit platform DOS) ?

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

int main()
{
    char *str1 = "India";
    char *str2 = "BIX";
    char *str3;
    str3 = strcat(str1, str2);
    printf("%s %s\n", str3, str1);
    return 0;
}
```

A. IndiaBIX India

B. IndiaBIX IndiaBIX

C. India India

D. Error

Answer: Option **B**
Explanation:

It prints 'IndiaBIX IndiaBIX' in TurboC (in 16 bit platform).

It may cause a 'segmentation fault error' in GCC (32 bit platform).

View Answer Discuss in Forum Workspace Report

---

28. If the size of pointer is 4 bytes then What will be the output of the program ?

```
#include<stdio.h>

int main()
{
    char *str[] = {"Frogs", "Do", "Not", "Die", "They", "Croak!"};
    printf("%d, %d", sizeof(str), strlen(str[0]));
    return 0;
}
```

A. 22, 4

B. 25, 5

C. 24, 5

D. 20, 2

Answer: Option **C**
Explanation:
Step 1: `char *str[] = {"Frogs", "Do", "Not", "Die", "They", "Croak!"};` The variable str is declared as an pointer to the array of 6 strings.
Step 2: `printf("%d, %d", sizeof(str), strlen(str[0]));`
`sizeof(str)` denotes 6 * 4 bytes = 24 bytes. Hence it prints '24'
`strlen(str[0]));` becomes `strlen(Frogs))`. Hence it prints '5';

Hence the output of the program is 24, 5

Hint: If you run the above code in 16 bit platform (Turbo C under DOS) the output will be 12, 5. Because the pointer occupies only 2 bytes. If you run the above code in Linux (32 bit platform), the output will be 24, 5 (because the size of pointer is 4 bytes).

View Answer Discuss in Forum Workspace Report

---

29. What will be the output of the program ?

```
#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: Option **B**
Explanation:

The function printf() returns the number of charecters printed on the console.

Step 1: `char a[] = '\0';` The variable `a` is declared as an array of characters and it initialized with "\0". It denotes that the string is empty.
Step 2: `if(printf("%s", a))` The `printf()` statement does not print anything, so it returns '0'(zero). Hence the if condition is failed.
In the `else` part it prints "The string is empty".
View Answer Discuss in Forum Workspace Report

30. What will be the output of the program ?

```
#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: Option **A**
Explanation:
`gets()` gets collects a string of characters terminated by a new line from the standard input stream `stdin`.
The `gets(str1)` read the input string from user and store in variable `str1`.
The `gets(str2)` read the input string from user and store in variable `str2`.
The code `i = strcmp(str1, str2);` The strcmp not only returns -1, 0 and +1, but also other negative or positive values. So the value of `i` is "unpredictable integer value".
`printf("%d\n", i);` It prints the value of variable `i`.

31. What will be the output of the program in Turbo C?

```
#include<stdio.h>

int main()
{
    char str[10] = "India";
    str[6] = "BIX";
    printf("%s\n", str);
    return 0;
}
```

A. India BIX

B. BIX

C. India

D. Error

Answer: Option **D**

Explanation:

`str[6] = "BIX";` - Nonportable pointer conversion.

View Answer Discuss in Forum Workspace Report

32. What will be the output of the program ?

```
#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 above

Answer: Option **B**

Explanation:

Step 1: `char str1[] = "Hello";` The variable `str1` is declared as an array of characters and initialized with a string "Hello".

Step 2: `char str2[] = "Hello";` The variable `str2` is declared as an array of characters and initialized with a string "Hello".

We have use `strcmp(s1,s2)` function to compare strings.

Step 3: `if(str1 == str2)` here the address of `str1` and `str2` are compared. The address of both variable is not same. Hence the if condition is failed.
Step 4: At the else part it prints "Unequal".
View Answer Discuss in Forum Workspace Report

---

33. What will be the output of the program ?

```
#include<stdio.h>

int main()
{
    char t;
    char *p1 = "India", *p2;
    p2=p1;
    p1 = "BIX";
    printf("%s %s\n", p1, p2);
    return 0;
}
```

A. India BIX

B. BIX India

C. India India

D. BIX BIX

Answer: Option **B**
Explanation:
Step 1: `char *p1 = "India", *p2;` The variable `p1` and `p2` is declared as an pointer to a character value and `p1` is assigned with a value "India".
Step 2: `p2=p1;` The value of `p1` is assigned to variable `p2`. So `p2` contains "India".
Step 3: `p1 = "BIX";` The `p1` is assigned with a string "BIX"
Step 4: `printf("%s %s\n", p1, p2);` It prints the value of `p1` and `p2`.

Hence the output of the program is "BIX India".

View Answer Discuss in Forum Workspace Report

---

34. What will be the output of the program ?

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

int main()
{
    printf("%c\n", "abcdefgh"[4]);
    return 0;
}
```

A. Error

B. d

C. e

D. abcdefgh

Answer: Option **C**
Explanation:
`printf("%c\n", "abcdefgh"[4]);` It prints the 5 character of the string "abcdefgh".

Hence the output is 'e'.

View Answer Discuss in Forum Workspace Report

---

35. What will be the output of the following program in 16 bit platform assuming that 1022 is memory address of the string "Hello1" (in Turbo C under DOS) ?

```
#include<stdio.h>

int main()
{
    printf("%u %s\n", &"Hello1", &"Hello2");
    return 0;
}
```

A. 1022 Hello2

B. Hello1 1022

C. Hello1 Hello2

D. 1022 1022

E. Error

Answer: Option **A**
Explanation:
In `printf("%u %s\n", &"Hello", &"Hello");`.
The `%u` format specifier tells the compiler to print the memory address of the `"Hello1"`.
The `%s` format specifier tells the compiler to print the string `"Hello2"`.

Hence the output of the program is "1022 Hello2".

31. What will be the output of the program in Turbo C?

```
#include<stdio.h>

int main()
{
    char str[10] = "India";
    str[6] = "BIX";
    printf("%s\n", str);
    return 0;
}
```

A. India BIX

B. BIX

C. India

D. Error

Answer: Option **D**

Explanation:

`str[6] = "BIX";` - Nonportable pointer conversion.

View Answer Discuss in Forum Workspace Report

32. What will be the output of the program ?

```
#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 above

Answer: Option **B**

Explanation:

Step 1: `char str1[] = "Hello";` The variable `str1` is declared as an array of characters and initialized with a string "Hello".

Step 2: `char str2[] = "Hello";` The variable `str2` is declared as an array of characters and initialized with a string "Hello".

We have use `strcmp(s1,s2)` function to compare strings.

Step 3: `if(str1 == str2)` here the address of `str1` and `str2` are compared. The address of both variable is not same. Hence the if condition is failed.

Step 4: At the else part it prints "Unequal".

View Answer Discuss in Forum Workspace Report

33. What will be the output of the program ?

```
#include<stdio.h>

int main()
{
    char t;
    char *p1 = "India", *p2;
    p2=p1;
    p1 = "BIX";
    printf("%s %s\n", p1, p2);
    return 0;
}
```

A. India BIX

B. BIX India

C. India India

D. BIX BIX

Answer: Option **B**

Explanation:

Step 1: `char *p1 = "India", *p2;` The variable `p1` and `p2` is declared as an pointer to a character value and `p1` is assigned with a value "India".
Step 2: `p2=p1;` The value of `p1` is assigned to variable `p2`. So `p2` contains "India".
Step 3: `p1 = "BIX";` The `p1` is assigned with a string "BIX"
Step 4: `printf("%s %s\n", p1, p2);` It prints the value of `p1` and `p2`.

Hence the output of the program is "BIX India".

View Answer Discuss in Forum Workspace Report

34. What will be the output of the program ?

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

int main()
{
    printf("%c\n", "abcdefgh"[4]);
    return 0;
}
```

A. Error

B. d

C. e

D. abcdefgh

Answer: Option **C**

Explanation:

`printf("%c\n", "abcdefgh"[4]);` It prints the 5 character of the string "abcdefgh".

Hence the output is 'e'.

View Answer Discuss in Forum Workspace Report

35. What will be the output of the following program in 16 bit platform assuming that 1022 is memory address of the string "Hello1" (in Turbo C under DOS) ?

```
#include<stdio.h>

int main()
{
    printf("%u %s\n", &"Hello1", &"Hello2");
    return 0;
}
```

A. 1022 Hello2

B. Hello1 1022

C. Hello1 Hello2

D. 1022 1022

E. Error

Answer: Option **A**
Explanation:
In `printf("%u %s\n", &"Hello", &"Hello");`.
The `%u` format specifier tells the compiler to print the memory address of the `"Hello1"`.
The `%s` format specifier tells the compiler to print the string `"Hello2"`.

Hence the output of the program is "1022 Hello2".