



Talent Transformation (2019)

Home ► My courses ► Talent Transformation ► ttc2019_2 ► TCS ONLINE TEST - 2018 ► SET - 12 (Programming Concept)

Started on Wednesday, 22 August 2018, 12:11 AM

State Finished

Completed on Wednesday, 22 August 2018, 12:17 AM

Time taken 6 mins 28 secs

Grade 7.00 out of 10.00 (70%)

Question 1

Incorrect

Mark 0.00 out of 1.00

Flag question

Does this mentioning array name gives the base address in all the contexts?

Select one:

- ☒ a. Yes ✖
- ☐ b. No

Explanation:

No, Mentioning the array name in C or C++ gives the base address in all contexts except one.

Syntactically, the compiler treats the array name as a pointer to the first element. You can reference elements using array syntax, `a[n]`, or using pointer syntax, `*(a+n)`, and you can even mix the usages within an expression.

When you pass an array name as a function argument, you are passing the "value of the pointer", which means that you are implicitly passing the array by reference, even though all parameters in functions are "call by value".

The correct answer is: No

Question 2

Correct

Mark 1.00 out of 1.00

Flag question

What does the following declaration mean?

`int (*ptr)[10];`

Select one:

- ☐ a. ptr is an pointer to array
- ☐ b. ptr is an array of 10 integers
- ☒ c. ptr is a pointer to an array of 10 integers ✔
- ☐ d. ptr is array of pointers to 10 integers

The correct answer is: ptr is a pointer to an array of 10 integers

Question 3

Correct

Mark 1.00 out of 1.00

🚩 Flag question

What will be the output of the program in Turb C (under DOS)?

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

```
int main()
```

```
{
```

```
int arr[5], i=0;
```

```
while(i<5)
```

```
arr[i]=++i;
```

```
for(i=0; i<5; i++)
```

```
printf("%d, ", arr[i]);
```

```
return 0;
```

```
}
```

Select one:

- ☐ a. 0, 1, 2, 3, 4,
- ☒ b. Garbage value, 1, 2, 3, 4, ✓
- ☐ c. 1, 2, 3, 4, 5,
- ☐ d. 2, 3, 4, 5, 6,

Explanation:

Since C is a compiler dependent language, it may give different outputs at different platforms. We have given the TurboC Compiler (Windows) output.

Please try the above programs in Windows (Turbo-C Compiler) and Linux (GCC Compiler), you will understand the difference better.

The correct answer is: Garbage value, 1, 2, 3, 4,

Question 4

Correct

Mark 1.00 out of 1.00

🚩 Flag question

Which of the following statements are correct about 6 used in the program?

```
int num[6];
```

```
num[6]=21;
```

Select one:

- ☐ a. In the first statement 6 specifies a particular element, whereas in the second statement it specifies a type.
- ☐ b. In the first statement 6 specifies a particular element, whereas in the second statement it specifies a array size.
- ☐ c. In both the statement 6 specifies array size.

- ☒ d. In the first statement 6 specifies a array size, whereas in the second statement it specifies a particular element of array. ✓

Explanation:

The statement 'B' is correct, because `int num[6];` specifies the size of array and `num[6]=21;` designates the particular element(7th element) of the array.

The correct answer is: In the first statement 6 specifies a array size, whereas in the second statement it specifies a particular element of array.

Question 5

Correct

Mark 1.00 out of 1.00

Flag question

What will be the output of the program ?

```
#include <stdio.h >
int main()
{
    static int a[2][2] = {1, 2, 3, 4};
    int i, j;
    static int *p[] = {(int*)a, (int*)a+1, (int*)a+2};
    for(i=0; i<2; i++)
    {
        for(j=0; j<2; j++)
        {
            printf("%d, %d, %d, %d\n", *(p+i+j), *(j+p+i),
            *(i+p+j), *(p+j+i));
        }
    }
    return 0;
}
```

Select one:

- ☒ a. 1, 1, 1, 1
2, 2, 2, 2
2, 2, 2, 2
3, 3, 3, 3 ✓
- ☐ b. 1, 1, 1, 1
2, 3, 2, 3
3, 2, 3, 2
4, 4, 4, 4
- ☐ c. 1, 2, 3, 4
2, 3, 4, 1
3, 4, 1, 2
4, 1, 2, 3
- ☐ d. 1, 2, 1, 2
2, 3, 2, 3
3, 4, 3, 4
4, 2, 4, 2

The correct answer is: 1, 1, 1, 1
2, 2, 2, 2
2, 2, 2, 2
3, 3, 3, 3

Question 6

Correct

Mark 1.00 out of
1.00

Flag question

Assuming a integer 2-bytes, What will be the output of the program?

```
#include<stdio.h>
int main()
{
printf("%x\n", -1<<3);
return 0;
}
```

Select one:

- ☐ a. 0
- ☐ b. ffff
- ☐ c. -1
- ☒ d. fff8 ✓

Explanation:

The system will treat negative numbers in 2's complement metho

Example:

Assume the size of int is 2-bytes(16 bits). The integer value 1 is represented as given below:

Binary of 1: 00000000 00000001 (this is for positive value of 1)

1's complement of binary 1: 11111111 11111110

2's complement of binary 1: 11111111 11111111

Thy system will store '11111111 11111111' in memory to represent '-1'.

If we do left shift (3 bits) on 11111111 11111111 it will become as given below:

11111111 11111111 --- (left shift 3 times) ---> 11111111 11111000.

So, 11111111 11111000 --- (binary to hex) ---> FF F8. (Required Answer)

Note:

How is the negative number obtained from 2's complement value?

As stated above, -1 is represented as '11111111 11111111' in memory.

So, the system will take 2's complement of '11111111 11111111' to the get the original negative value back.

Example:

Bit Representation of -1: 11111111 11111111

Since the left most bit is 1, it is a negative number. Then the value is

1's complement: 00000000 00000000

2's complement: 00000000 00000001 (Add 1 to the above result)

Therefore, '00000000 00000001' = 1 and the sign is negative.
Hence the value is -1.

The correct answer is: fff8

Question 7

Correct

Mark 1.00 out of
1.00

🚩 Flag question

Which of the following statements are correct about the program?

```
#include<stdio.h>
int main()
{
    unsigned int num;
    int c=0;
    scanf("%u", &num);
    for(;num;num>>=1)
    {
        if(num & 1)
            c++;
    }
    printf("%d", c);
    return 0;
}
```

Select one:

- ☐ a. It counts the number of bits that are OFF (0) in the number num.
- ☐ b. Error
- ☐ c. It sets all bits in the number num to 1
- ☒ d. It counts the number of bits that are ON (1) in the number num. ✓

Explanation:

If we give input 4, it will print 1.

Binary-4 == 00000000 00000100 ; Total number of bits = 1.

If we give input 3, it will print 2.

Binary-3 == 00000000 00000011 ; Total number of bits = 2.

If we give input 511, it will print 9.

Binary-511 == 00000001 11111111 ; Total number of bits = 9.

The correct answer is: It counts the number of bits that are ON (1) in the number num.

Question 8

Incorrect

Mark 0.00 out of
1.00

🚩 Flag question

Which of the following statements are correct about the program?

```
#include<stdio.h>
int main()
{
    unsigned int num;
    int i;
```

```
scanf("%u", &num);
for(i=0; i<16; i++)
{
printf("%d", (num<<i & 1<<15)?1:0);
}
return 0;
}
```

Select one:

- ☐ a. It prints binary equivalent num
- ☒ b. It prints all even bits from num ❌
- ☐ c. It prints all odd bits from num
- ☐ d. Error

Explanation:

If we give input 4, it will print 00000000 00000100 ;

If we give input 3, it will print 00000000 00000011 ;

If we give input 511, it will print 00000001 11111111 ;

The correct answer is: It prints binary equivalent num

Question 9

Correct

Mark 1.00 out of
1.00

🚩 Flag question

How will you free the memory allocated by the following program?

```
#include<stdio.h>
#include<stdlib.h>
#define MAXROW 3
#define MAXCOL 4
int main()
{
int **p, i, j;
p = (int **) malloc(MAXROW * sizeof(int*));
return 0;
}
```


Select one:

- ☐ a. malloc(p, 0);
- ☒ b. free(p); ✓
- ☐ c. dealloc(p);
- ☐ d. memfree(int p);

The correct answer is: free(p);

Question 10


Incorrect

Mark 0.00 out of
1.00 Flag question

What will be the output of the program?

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
    int *p;
    p = (int *)malloc(20); /* Assume p has address of 1314 */
    free(p);
    printf("%u", p);
    return 0;
}
```

Select one:

- ☐ a. 1316
- ☒ b. Random address 
- ☐ c. 1314
- ☐ d. Garbage value

The correct answer is: 1314

[Finish review](#)**QUIZ NAVIGATION**

1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	----

[Show one page at a time](#)[Finish review](#)

You are logged in as JEET SAHA 16900215019 ([Log out](#))
ttc2019_2