Talent Transformation (2019)

Started on	Thursday, 23 August 2018, 12:10 AM
State	Finished
Completed on	Thursday, 23 August 2018, 12:16 AM
Time taken	5 mins 47 secs
Grade	**7.00** out of 10.00 (**70**%)

Question 1

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 array of pointers to 10 integers
- c. ptr is a pointer to an array of 10 integers ✓
- d. ptr is an array of 10 integers

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

Question 2

Incorrect

Mark 0.00 out of 1.00

Flag question

What will happen if in a C program you assign a value to an array element whose subscript exceeds the size of array?

Select one:

- a. The compiler would report an error. ✗
- b. The element will be set to 0.
- c. The array size would appropriately grow.
- d. The program may crash if some important data gets overwritten.

Explanation:

If the index of the array size is exceeded, the program will crash. Hence "option c" is the correct answer. But the modern compilers will take care of this kind of errors.

Example: Run the below program, it will crash in Windows (TurboC Compiler)

```
#include<stdio.h>
int main()
{
int arr[2];
arr[3]=10;
printf("%d",arr[3]);
return 0;
}
```

Since C is a compiler dependent language, it may give different outputs at different platforms. We have given the Turbo-C 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: The program may crash if some important data gets overwritten.

Question 3

Incorrect

Mark 0.00 out of 1.00

Flag question

Is there any difference int the following declarations?

```
int fun(int arr[]);
int fun(int arr[2]);
```

Select one:

- a. Yes ✗
- b. No

Explanation:

No, both the statements are same. It is the prototype for the function fun() that accepts one integer array as an parameter and returns an integer value.

The correct answer is: No

Question 4

Correct

Mark 1.00 out of 1.00

Flag question

What will be the output of the program if the array begins 1200 in memory?

```
#include<stdio.h>
int main()
{
int arr[]={2, 3, 4, 1, 6};
printf("%u, %u, %u\n", arr, &arr[0], &arr);
return 0;
}
```

Select one:

- a. 1200, 1202, 1200
- b. 1200, 1204, 1208
- c. 1200, 1202, 1204

d. 1200, 1200, 1200 ✓

Explanation:

Step 1: int arr[]={2, 3, 4, 1, 6}; The variable arr is declared as an integer array and initialize
Step 2: printf("%u, %u, %u\n", arr, &arr[0], &arr); Here,
The base address of the array is 1200.
=> arr, &arr is pointing to the base address of the array arr.
=> &arr[0] is pointing to the address of the first element array arr. (ie. base address)
Hence the output of the program is 1200, 1200, 1200

The correct answer is: 1200, 1200, 1200

Question 5

Correct

Mark 1.00 out of 1.00

Flag question

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

Select one:

a. No ✓

b. Yes

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 6

Correct

Mark 1.00 out of 1.00

Flag question

What will be the output of the program ?

```
#include<stdio.h>
int main()
{
void *vp;
char ch=74, *cp="JACK";
int j=65;
vp=&ch;
printf("%c", *(char*)vp);
vp=&j;
```

```
printf("%c", *(int*)vp);
vp=cp;
printf("%s", (char*)vp+2);
return 0;
}
```

Select one:

- a. J65K
- b. JAK
- c. JCK
- d. JACK ✓

The correct answer is: JACK

Question 7

Correct

Mark 1.00 out of 1.00

Flag question

In the following program add a statement in the function fun() such that address ofa gets stored in j?

```
#include<stdio.h>
int main()
{
int *j;
void fun(int**);
fun(&j);
return 0;
}
void fun(int **k)
{
int a=10;
/* Add a statement here */
}
```

Select one:

- a. **k=a;
- b. *k=&a ✓
- c. k=&a;
- d. &k=*a

The correct answer is: *k=&a

Question 8

Correct

What will be the output of the program ?

```
#include<stdio.h>
void fun(void *p);
int i;
```

Mark 1.00 out of 1.00

Flag question

```
int main()
{
void *vptr;
vptr = &i;
fun(vptr);
return 0;
}
void fun(void *p)
{
int **q;
q = (int**)&p;
printf("%d\n", **q);
}
```

Select one:

- a. Garbage value
- b. 0 ✓
- c. No output
- d. Error: cannot convert from void** to int**

The correct answer is: 0

Question 9

Correct

Mark 1.00 out of 1.00

Flag question

Are the expression *ptr++ and ++*ptr are same?

Select one:

- a. True
- b. False ✓

Explanation:

*ptr++ increments the pointer and not the value, whereas the ++*ptr increments the value being pointed by ptr

The correct answer is: False

Question 10

Incorrect

Mark 0.00 out of 1.00

Flag question

Can you combine the following two statements into one?

```
char *p;
p = (char*) malloc(100);
```

Select one:

- a. char *p = (char*)malloc(100);
- b. char *p = (char) malloc(100); ✗

c. char *p = (char *)(malloc*)(100);

d. char p = *malloc(100);

The correct answer is: char *p = (char*)malloc(100);

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