Talent Transformation (2019)

Started on	Tuesday, 21 August 2018, 11:29 PM
State	Finished
Completed on	Tuesday, 21 August 2018, 11:36 PM
Time taken	7 mins 20 secs
Grade	**6.00** out of 10.00 (**60**%)

Question 1

Correct

Mark 1.00 out of 1.00

Flag question

What will be the output of the program?

```
#include<stdio.h>
int main()
{
int x=12, y=7, z;
z = x!=4 || y == 2;
printf("z=%d\n", z);
return 0;
}
```

Select one:

- a. z=2
- b. z=4
- c. z=0
- d. z=1 ✓

Explanation:

Step 1: int x=12, y=7, z; here variable x, y and z are declared as an integer and variable x and y are initialized to 12, 7 respectively.
Step 2: z = x!=4 || y == 2;
becomes z = 12!=4 || 7 == 2;
then z = (condition true) || (condition false); Hence it returns 1. So the value of z=1.
Step 3: printf("z=%d\n", z); Hence the output of the program is "z=1".

The correct answer is: z=1

Question 2

Correct

Mark 1.00 out of 1.00

Flag question

What will be the output of the program?

```
#include<stdio.h>
int main()
{
int i=-3, j=2, k=0, m;
m = ++i || ++j && ++k;
printf("%d, %d, %d, %d\n", i, j, k, m);
return 0;
}
```

Select one:

- a. 2, 2, 0, 1
- b. 1, 2, 1, 0
- c. -2, 2, 0, 0
- d. -2, 2, 0, 1 ✓

Explanation:

Step 1: int i=-3, j=2, k=0, m; here variable i, j, k, m are declared as an integer type and variable i, j, k are initialized to -3, 2, 0 respectively.
Step 2: m = ++i || ++j && ++k; here (++j && ++k;) this code will not get executed because ++i has non-zero value.
becomes m = -2 || ++j && ++k;
becomes m = TRUE || ++j && ++k; Hence this statement becomes TRUE. So it returns '1'(one). Hence m=1.
Step 3: printf("%d, %d, %d, %d\n", i, j, k, m); In the previous step the value of variable 'i' only increemented by '1'(one). The variable j,k are not increemente
Hence the output is "-2, 2, 0, 1".

The correct answer is: -2, 2, 0, 1

Question 3

Incorrect

Mark 0.00 out of 1.00

Flag question

What will be the output of the program?

```
#include<stdio.h>
int main()
{
int i=2;
printf("%d, %d\n", ++i, ++i);
return 0;
}
```

Select one:

- a. 4, 4
- b. 3, 4 ✗
- c. 4, 3

d. Output may vary from compiler to compiler

Explanation:

The order of evaluation of arguments passed to a function call is unspecifie
Anyhow, we consider ++i, ++i are Right-to-Left associativity. The output of the program is 4, 3.
In TurboC, the output will be 4, 3.
In GCC, the output will be 4, 4.

The correct answer is: Output may vary from compiler to compiler

Question 4

Correct

Mark 1.00 out of 1.00

Flag question

Will the program compile successfully?

```
#include<stdio.h>
int main()
{
#ifdef NOTE
int a;
a=10;
#else
int a;
a=20;
#endif
printf("%d\n", a);
return 0;
}
```

Select one:

a. No

b. Yes ✓

Explanation:

Yes, this program will compile and run successfully and prints 20.
The macro #ifdef NOTE evaluates the given expression to 1. If satisfied it executes the #ifdef block statements. Here #ifdef condition fails because the Macro NOTE is nowhere declared.
Hence the #else block gets executed, the variable a is declared and assigned a value of 20.
printf("%d\n", a); It prints the value of variable a 20.

The correct answer is: Yes

Question 5

Incorrect

Mark 0.00 out of 1.00

Flag question

Preprocessor directive #ifdef .. #else ... #endif is used for conditional compilation.

Select one:

- a. False ✗
- b. True

Explanation:

True, these macros are used for conditional operation.
#if <constant-expression>
#elif <constant-expression>
#endif

The correct answer is: True

Question 6

Incorrect

Mark 0.00 out of 1.00

Flag question

What will be the output of the program?

```
#include<stdio.h>
#define MAN(x, y) ((x)>(y)) ? (x):(y);
int main()
{
int i=10, j=5, k=0;
k = MAN(++i, j++);
printf("%d, %d, %d\n", i, j, k);
return 0;
}
```

Select one:

- a. 12, 6, Garbage
- b. 12, 6, 12
- c. 11, 5, 11 ✗
- d. 11, 5, Garbage

Explanation:

The macro MAN(x, y) ((x)>(y)) ? (x):(y); returns the biggest number of given two numbers.
Step 1: int i=10, j=5, k=0; The variable i, j, k are declared as an integer type and initialized to value 10, 5, 0 respectively.
Step 2: k = MAN(++i, j++); becomes,
=> k = ((++i)>(j++)) ? (++i):(j++);
=> k = ((11)>(5)) ? (12):(6);
=> k = 12

Step 3: printf("%d, %d, %d\n", i, j, k); It prints the variable i, j, k.
In the above macro step 2 the variable i value is increemented by 2 and variable jvalue is increemented by 1.
Hence the output of the program is 12, 6, 12

The correct answer is: 12, 6, 12

Question 7

Incorrect

Mark 0.00 out of 1.00

Flag question

Point out the correct statement which correctly free the memory pointed to by 's' and 'p' in the following program?

```
#include<stdio.h>
#include<stdlih>
int main()
{
struct ex
{
int i;
float j;
char *s
};
struct ex *p;
p = (struct ex *)malloc(sizeof(struct ex));
p->s = (char*)malloc(20);
return 0;
}
```

Select one:

- a. free(p); ✗
- b. free(p->s);
- c. free(p); , free(p->s);
- d. free(p->s); , free(p);

The correct answer is: free(p->s); , free(p);

Question 8

Correct

Mark 1.00 out of 1.00

Flag question

Assume integer is 2 bytes wide. What will be the output of the following code?

```
#include<stdio.h>
#include<stdlih>
#define MAXROW 3
#define MAXCOL 4
int main()
{
int (*p)[MAXCOL];
p = (int (*) [MAXCOL])malloc(MAXROW *sizeof(*p));
printf("%d, %d\n", sizeof(p), sizeof(*p));
return 0;
}
```

Select one:

- a. 4, 16
- b. 16, 32
- c. 2, 8 ✓
- d. 8, 24

The correct answer is: 2, 8

Question 9

Correct

Mark 1.00 out of 1.00

Flag question

Every time we supply new set of values to the program at command prompt, we need to recompile the program.

Select one:

- a. True
- b. False ✓

The correct answer is: False

Question 10

Correct

Mark 1.00 out of 1.00

Flag question

A pointer to a block of memory is effectively same as an array

Select one:

- a. False
- b. True ✓

Explanation:

Yes, It is possible to allocate a block of memory (of arbitrary size) at run-time, using the standard library's malloc function, and treat it as an array.

The correct answer is: True

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