Search...

Online C Programming Test :: C Programming Test 8

Marks : 15/20		
Total number of questions	:	**20**
Number of answered questions	:	**19**
Number of unanswered questions	:	**1**

Test Review : View answers and explanation for this test.

1. Which of the following cannot be checked in a *switch-case* statement?
   - A. Character ✖
   - B. Integer ✖
   - C. Float ✔
   - D. enum ✖

Your Answer: Option C

Correct Answer: Option C

Explanation:

The *switch/case* statement in the c language is defined by the language specification to use an *int* value, so you can not use a *float* value.

```
switch( expression )
{
    case constant-expression1:    statements 1;
    case constant-expression2:    statements 2;
    case constant-expression3:    statements3 ;
    ...
    ...
    default : statements 4;
}
```

The value of the '*expression*' in a switch-case statement must be an integer, char, short, long. Float and double are not allowed.

Learn more problems on : Control Instructions

Discuss about this problem : Discuss in Forum

2. Point out the error, if any in the program.

```
#include<stdio.h>
int main()
{
    int a = 10, b;
    a >=5 ? b=100: b=200;
    printf("%d\n", b);
    return 0;
}
```

- A. 100 ✗
- B. 200 ✗
- C. Error: L value required for b ✓
- D. Garbage value ✗

Your Answer: Option C

Correct Answer: Option C

Explanation:

Variable b is not assigned.

It should be like:

b = a >= 5 ? 100 : 200;

Learn more problems on : Control Instructions

Discuss about this problem : Discuss in Forum

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

```
#include<stdio.h>
int main()
{
    int x = 30, y = 40;
    if(x == y)
        printf("x is equal to y\n");

    else if(x > y)
        printf("x is greater than y\n");

    else if(x < y)
        printf("x is less than y\n")
    return 0;
}
```

- A. Error: Statement missing ✓
- B. Error: Expression syntax ✗
- C. Error: Lvalue required ✗
- D. Error: Rvalue required ✗

Your Answer: Option (Not Answered)

Correct Answer: Option A

Explanation:

This program will result in error "Statement missing ;"

printf("x is less than y\n") here ; should be added to the end of this statement.

Learn more problems on : Control Instructions

Discuss about this problem : Discuss in Forum

4. What will be the output of the program?

```
#include<stdio.h>
int main()
{
    int i=4, j=-1, k=0, w, x, y, z;
    w = i || j || k;
    x = i && j && k;
    y = i || j &&k;
    z = i && j || k;
    printf("%d, %d, %d, %d\n", w, x, y, z);
    return 0;
}
```

- A. 1, 1, 1, 1 ✗
- B. 1, 1, 0, 1 ✗
- C. 1, 0, 0, 1 ✗
- D. 1, 0, 1, 1 ✓

Your Answer: Option D

Correct Answer: Option D

Explanation:

Step 1: *int i=4, j=-1, k=0, w, x, y, z;* here variable *i, j, k, w, x, y, z* are declared as an integer type and the variable *i, j, k* are initialized to 4, -1, 0 respectively.

Step 2: *w = i || j || k;* becomes *w = 4 || -1 || 0;*. Hence it returns TRUE. So, w=1

Step 3: *x = i && j && k;* becomes *x = 4 && -1 && 0;* Hence it returns FALSE. So, x=0

Step 4: *y = i || j &&k;* becomes *y = 4 || -1 && 0;* Hence it returns TRUE. So, y=1

Step 5: *z = i && j || k;* becomes *z = 4 && -1 || 0;* Hence it returns TRUE. So, z=1.

Step 6: *printf("%d, %d, %d, %d\n", w, x, y, z);* Hence the output is "1, 0, 1, 1".

Learn more problems on : Expressions

Discuss about this problem : Discuss in Forum

5. Are the following two statement same?
   1. a <= 20 ? (b = 30): (c = 30);
   2. (a <=20) ? b : (c = 30);

- A. Yes ✗
- B. No ✓

Your Answer: Option B

Correct Answer: Option B

Explanation:

No, the expressions 1 and 2 are not same.

1. *a <= 20 ? (b = 30) : (c = 30);* This statement can be rewritten as,

```
if(a <= 20)
{
    b = 30;
}
else
{
    c = 30;
}
```

2. *(a <=20) ? b : (c = 30);* This statement can be rewritten as,

```
if(a <= 20)
{
    //Nothing here
}
else
{
    c = 30;
}
```

Learn more problems on : Expressions

Discuss about this problem : Discuss in Forum

6.Macros have a local scope.

- A.True ✖
- B.False ✔

Your Answer: Option A

Correct Answer: Option B

Explanation:

False, The scope of macros is globals and functions. Also the scope of macros is only from the point of definition to the end of the file.

Learn more problems on : C Preprocessor

Discuss about this problem : Discuss in Forum

7. What will be the output of the program ?

```
#include<stdio.h>

int main()
{
    char *str;
    str = "%d\n";
```

```
    str++;
    str++;
    printf(str-2, 300);
    return 0;
}
```

- A.No output ✗
- B.30 ✗
- C.3 ✗
- D.300 ✓

Your Answer: Option D

Correct Answer: Option D

Learn more problems on : Pointers

Discuss about this problem : Discuss in Forum

8. Is this a correct way for NULL pointer assignment?
int i=0;
*char *q=(char*)i;*
- A.Yes ✗
- B.No ✓

Your Answer: Option B

Correct Answer: Option B

Explanation:

The correct way is *char *q=0 (or) char *q=(char*)0*

Learn more problems on : Pointers

Discuss about this problem : Discuss in Forum

9.What will be the output of the program if the array begins at 65472 and each integer occupies 2 bytes?

```
#include<stdio.h>

int main()
{
    int a[3][4] = {1, 2, 3, 4, 4, 3, 2, 1, 7, 8, 9, 0};
    printf("%u, %u\n", a+1, &a+1);
    return 0;
}
```

- A.65474, 65476 ✗
- B.65480, 65496 ✓
- C.65480, 65488 ✗
- D.65474, 65488 ✗

Your Answer: Option D

Correct Answer: Option B

Explanation:

Step 1: *int a[3][4] = {1, 2, 3, 4, 4, 3, 2, 1, 7, 8, 9, 0};* The array a[3][4] is declared as an integer array having the 3 rows and 4 colums dimensions.

Step 2: *printf("%u, %u\n", a+1, &a+1);*

The base address(also the address of the first element) of array is 65472.

For a two-dimensional array like *a* reference to array has type "pointer to array of 4 ints". Therefore, *a+1* is pointing to the memory location of first element of the second row in array *a*. Hence 65472 + (4 ints * 2 bytes) = 65480

Then, *&a* has type "pointer to array of 3 arrays of 4 ints", totally 12 ints. Therefore, *&a+1* denotes "12 ints * 2 bytes * 1 = 24 bytes".

Hence, begining address 65472 + 24 = 65496. So, *&a+1* = 65496

Hence the output of the program is 65480, 65496

Learn more problems on : Arrays

Discuss about this problem : Discuss in Forum

10. The library function used to reverse a string is

- A.strstr() ✗
- B.strrev() ✓
- C.revstr() ✗
- D.strreverse() ✗

Your Answer: Option B

Correct Answer: Option B

Explanation:

strrev(s) Reverses all characters in *s*

Example:

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

int main(void)
{
   char *str = "IndiaBIX";

   printf("Before strrev(): %s\n", str);
   strrev(str);
   printf("After strrev():  %s\n", str);
   return 0;
}
```

Output:

Before strrev(): IndiaBIX

After strrev(): XIBaidnI

Learn more problems on : Strings

Discuss about this problem : Discuss in Forum

11. 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 ✖

Your Answer: Option B

Correct Answer: Option B

Learn more problems on : Strings

Discuss about this problem : Discuss in Forum

12. What will be the output of the program ?

```
#include<stdio.h>

    struct course
    {
        int courseno;
        char coursename[25];
    };
int main()
{
    struct course c[] = { {102, "Java"},
                          {103, "PHP"},
                          {104, "DotNet"}     };

    printf("%d ", c[1].courseno);
    printf("%s\n", (*(c+2)).coursename);
    return 0;
}
```

- A. 103 DotNet ✔
- B. 102 Java ✖
- C. 103 PHP ✖
- D. 104 DotNet ✖

Your Answer: Option A

Correct Answer: Option A

Learn more problems on : Structures, Unions, Enums

Discuss about this problem : Discuss in Forum

13. Point out the error in the program?

```
struct emp
{
    int ecode;
    struct emp *e;
};
```

- A. Error: in structure declaration ✗
- B. Linker Error ✗
- C. No Error ✓
- D. None of above ✗

Your Answer: Option C

Correct Answer: Option C

Explanation:

This type of declaration is called as self-referential structure. Here **e* is pointer to a *struct emp*.

Learn more problems on : Structures, Unions, Enums

Discuss about this problem : Discuss in Forum

14. On declaring a structure 0 bytes are reserved in memory.
    - A. True ✗
    - B. False ✓

    Your Answer: Option B

    Correct Answer: Option B

    Learn more problems on : Structures, Unions, Enums

    Discuss about this problem : Discuss in Forum

15. On executing the below program what will be the contents of 'target.txt' file if the source file contains a line "To err is human"?

```
#include<stdio.h>

int main()
{
    int i, fss;
    char ch, source[20] = "source.txt", target[20]="target.txt", t;
    FILE *fs, *ft;
    fs = fopen(source, "r");
    ft = fopen(target, "w");
    while(1)
    {
        ch=getc(fs);
        if(ch==EOF)
            break;
        else
        {
            fseek(fs, 4L, SEEK_CUR);
            fputc(ch, ft);
        }
    }
    return 0;
}
```

- A.r n ✗
- B.Trh ✓
- C.err ✗
- D.None of above ✗

Your Answer: Option B

Correct Answer: Option B

Explanation:

The file *source.txt* is opened in read mode and *target.txt* is opened in write mode. The file *source.txt* contains "To err is human".

Inside the while loop,

ch=getc(fs); The first character('T') of the *source.txt* is stored in variable *ch* and it's checked for *EOF*.

if(ch==EOF) If *EOF*(End of file) is true, the loop breaks and program execution stops.

If not *EOF* encountered, *fseek(fs, 4L, SEEK_CUR);* the file pointer advances 4 character from the current position. Hence the file pointer is in 5th character of file *source.txt*.

fputc(ch, ft); It writes the character 'T' stored in variable *ch* to *target.txt*.

The *while* loop runs three times and it write the character 1st and 5th and 11th characters ("Trh") in the *target.txt* file.

Learn more problems on : Input / Output

Discuss about this problem : Discuss in Forum

16. What will be the output of the program (sample.c) given below if it is executed from the command line (Turbo C in DOS)?
cmd> *sample 1 2 3*

```
/* sample.c */
#include<stdio.h>

int main(int argc, char *argv[])
{
    int j;
    j = argv[1] + argv[2] + argv[3];
    printf("%d", j);
    return 0;
}
```

- A.6 ✗
- B.sample 6 ✗
- C.Error ✓
- D.Garbage value ✗

Your Answer: Option C

Correct Answer: Option C

Explanation:

Here *argv[1]*, *argv[2]* and *argv[3]* are string type. We have to convert the string to integer type before perform arithmetic operation.

Example: *j = atoi(argv[1]) + atoi(argv[2]) + atoi(argv[3]);*

Learn more problems on : Command Line Arguments

Discuss about this problem : Discuss in Forum

17. In Turbo C/C++ under DOS if we want that any wild card characters in the command-line arguments should be appropriately expanded, are we required to make any special provision?

- A. Yes ✓
- B. No ✗

Your Answer: Option B

Correct Answer: Option A

Explanation:

Yes you have to compile a program like
tcc myprog wildargs.obj

Learn more problems on : Command Line Arguments

Discuss about this problem : Discuss in Forum

18. Bitwise | can be used to multiply a number by powers of 2.

- A. Yes ✗
- B. No ✓

Your Answer: Option B

Correct Answer: Option B

Learn more problems on : Bitwise Operators

Discuss about this problem : Discuss in Forum

19. What do the following declaration signify?

```
int *f();
```

- A. *f* is a pointer variable of function type. ✗
- B. *f* is a function returning pointer to an *int*. ✓
- C. *f* is a function pointer. ✗
- D. *f* is a simple declaration of pointer variable. ✗

Your Answer: Option B

Correct Answer: Option B

Learn more problems on : Complicated Declarations

Discuss about this problem : Discuss in Forum

20. What will be the output of the program under DOS?

```
#include<stdio.h>

int main()
{
    char huge *near *far *ptr1;
    char near *far *huge *ptr2;
    char far *huge *near *ptr3;
    printf("%d, %d, %d\n", sizeof(ptr1), sizeof(**ptr2), sizeof(ptr3));
    return 0;
}
```

- A.4, 4, 4 ✗
- B.4, 2, 2 ✓
- C.2, 8, 4 ✗
- D.2, 4, 8 ✗

Your Answer: Option D

Correct Answer: Option B

Learn more problems on : Complicated Declarations

Discuss about this problem : Discuss in Forum

*** END OF THE TEST ***

Feedback

Quality of the Test : -- Select --

Difficulty of the Test : -- Select --

Comments:

. . .

Submit Feedback

Current Affairs 2018

Interview Questions and Answers