Online C Programming Test :: C Programming Test 7

Marks : 17/20

Total number of questions	:	**20**
Number of answered questions	:	**20**
Number of unanswered questions	:	**0**

Test Review : View answers and explanation for this test.

1. A *short integer* is at least 16 bits wide and a *long integer* is at least 32 bits wide.
   - [✓] A. True ✓
   - [ ] B. False ✗

   Your Answer: Option A

   Correct Answer: Option A

   Explanation:

   The basic C compiler is 16 bit compiler, below are the size of it's data types
   The size of *short int* is 2 bytes wide(16 bits).
   The size of *long int* is 4 bytes wide(32 bits).

   Learn more problems on : Control Instructions

   Discuss about this problem : Discuss in Forum

2. Which of the following correctly shows the hierarchy of arithmetic operations in C?
   - [ ] A. / + * - ✗
   - [ ] B. * - / + ✗
   - [ ] C. + - / * ✗
   - [✓] D. / * + - ✓

   Your Answer: Option D

Correct Answer: Option D

Explanation:

Simply called as BODMAS (Bracket of Division, Multiplication, Addition and Subtraction).

How Do I Remember ? BODMAS !

- **B** - Brackets first
- **O** - Orders (ie Powers and Square Roots, etc.)
- **DM** - Division and Multiplication (left-to-right)
- **AS** - Addition and Subtraction (left-to-right)

Learn more problems on : Expressions

Discuss about this problem : Discuss in Forum

3. In which order do the following gets evaluated
   1. Relational
   2. Arithmetic
   3. Logical
   4. Assignment

   [✓ A.2134] ✓
   B.1234 ✗
   C.4321 ✗
   D.3214 ✗

   Your Answer: Option A

   Correct Answer: Option A

   Explanation:

   2. Arithmetic operators: *, /, %, +, -
   1. Relational operators: >, <, >=, <=, ==, !=
   3. Logical operators : !, &&, ||
   4. Assignment operators: =

   Learn more problems on : Expressions

   Discuss about this problem : Discuss in Forum

4. Associativity has no role to play unless the precedence of operator is same.
   A.True ✓
   [✓ B.False] ✗

   Your Answer: Option B

   Correct Answer: Option A

   Explanation:

   Associativity is only needed when the operators in an expression have the same precedence. Usually + and - have the same precedence.

Consider the expression *7 - 4 + 2*. The result could be either *(7 - 4) + 2 = 5* or *7 - (4 + 2) = 1*. The former result corresponds to the case when + and - are left-associative, the latter to when + and - are right-associative.

Usually the addition, subtraction, multiplication, and division operators are left-associative, while the exponentiation, assignment and conditional operators are right-associative. To prevent cases where operands would be associated with two operators, or no operator at all, operators with the same precedence must have the same associativity.

Learn more problems on : Expressions

Discuss about this problem : Discuss in Forum

5.If the binary eauivalent of 5.375 in normalised form is 0100 0000 1010 1100 0000 0000 0000 0000, what will be the output of the program (on intel machine)?

```
#include<stdio.h>
#include<math.h>
int main()
{
    float a=5.375;
    char *p;
    int i;
    p = (char*)&a;
    for(i=0; i<=3; i++)
        printf("%02x\n", (unsigned char)p[i]);
    return 0;
}
```

- ☐ A.40 AC 00 00 ✗
- ☐ B.04 CA 00 00 ✗
- [✓ C.00 00 AC 40] ✓
- ☐ D.00 00 CA 04 ✗

Your Answer: Option C

Correct Answer: Option C

Learn more problems on : Floating Point Issues

Discuss about this problem : Discuss in Forum

6.A *float* occupies 4 bytes. If the hexadecimal equivalent of these 4 bytes are A, B, C and D, then when this *float* is stored in memory in which of the following order do these bytes gets stored?

- ☐ A.ABCD ✗
- ☐ B.DCBA ✗
- ☐ C.0xABCD ✗
- [✓ D.Depends on big endian or little endian architecture] ✓

Your Answer: Option D

Correct Answer: Option D

Learn more problems on : Floating Point Issues

Discuss about this problem : Discuss in Forum

7. What will be the output of the program?

```
#include<stdio.h>
int sumdig(int);
int main()
{
    int a, b;
    a = sumdig(123);
    b = sumdig(123);
    printf("%d, %d\n", a, b);
    return 0;
}
int sumdig(int n)
{
    int s, d;
    if(n!=0)
    {
        d = n%10;
        n = n/10;
        s = d+sumdig(n);
    }
    else
        return 0;
    return s;
}
```

- ☐ A. 4, 4 ✗
- ☐ B. 3, 3 ✗
- ☑ C. 6, 6 ✓
- ☐ D. 12, 12 ✗

Your Answer: Option C

Correct Answer: Option C

Learn more problems on : Functions

Discuss about this problem : Discuss in Forum

8. Point out the error in the program

```
f(int a, int b)
{
    int a;
    a = 20;
    return a;
}
```

- ☐ A. Missing parenthesis in *return* statement ✗
- ☑ B. The function should be defined as *int f(int a, int b)* ✗
- ☐ C. Redeclaration of *a* ✓
- ☐ D. None of above ✗

Your Answer: Option B

Correct Answer: Option C

Explanation:

f(int a, int b) The variable *a* is declared in the function argument statement.

int a; Here again we are declaring the variable *a*. Hence it shows the error "Redeclaration of a"

Learn more problems on : Functions

Discuss about this problem : Discuss in Forum

9.What will be the output of the program?

```
#include<stdio.h>
#define SQUARE(x) x*x

int main()
{
    float s=10, u=30, t=2, a;
    a = 2*(s-u*t)/SQUARE(t);
    printf("Result = %f", a);
    return 0;
}
```

[✓ A.Result = -100.000000] ✓
B.Result = -25.000000 ✗
C.Result = 0.000000 ✗
D.Result = 100.000000 ✗

Your Answer: Option A

Correct Answer: Option A

Explanation:

The macro function *SQUARE(x) x*x* calculate the square of the given number *'x'*. (Eg: 10^2)

Step 1: *float s=10, u=30, t=2, a;* Here the variable *s, u, t, a* are declared as an floating point type and the variable *s, u, t* are initialized to 10, 30, 2.

Step 2: *a = 2*(s-u*t)/SQUARE(t);* becomes,

=> *a = 2 * (10 - 30 * 2) / t * t;* Here *SQUARE(t)* is replaced by macro to *t*t* .

=> *a = 2 * (10 - 30 * 2) / 2 * 2;*

=> *a = 2 * (10 - 60) / 2 * 2;*

=> *a = 2 * (-50) / 2 * 2 ;*

=> *a = 2 * (-25) * 2 ;*

=> *a = (-50) * 2 ;*

=> *a = -100;*

Step 3: *printf("Result=%f", a);* It prints the value of variable 'a'.

Hence the output of the program is -100

Learn more problems on : C Preprocessor

Discuss about this problem : Discuss in Forum

10. Preprocessor directive *#undef* can be used only on a macro that has been *#define* earlier
[✓ A.True] ✓
B.False ✗

Your Answer: Option A

Correct Answer: Option A

Explanation:

True, *#undef* can be used only on a macro that has been *#define* earlier

Example: *#define PI 3.14*

We can undefine *PI* macro by *#undef PI*

Learn more problems on : C Preprocessor

Discuss about this problem : Discuss in Forum

11. Which statement will you add to the following program to ensure that the program outputs "IndiaBIX" on execution?

```
#include<stdio.h>

int main()
{
    char s[] = "IndiaBIX";
    char t[25];
    char *ps, *pt;
    ps = s;
    pt = t;
    while(*ps)
        *pt++ = *ps++;

    /* Add a statement here */
    printf("%s\n", t);
    return 0;
}
```

☐ A. *pt=''; ✗
☐ B. pt='\0'; ✗
☐ C. pt='\n'; ✗
☑ D. *pt='\0'; ✓

Your Answer: Option D

Correct Answer: Option D

Learn more problems on : Pointers

Discuss about this problem : Discuss in Forum

12. 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 ✗

Your Answer: Option C

Correct 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".

Learn more problems on : Strings

Discuss about this problem : Discuss in Forum

13.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 ✗

Your Answer: Option A

Correct 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'.

Learn more problems on : Strings

Discuss about this problem : Discuss in Forum

14.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 ✗

Your Answer: Option C

Correct 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

Learn more problems on : Strings

Discuss about this problem : Discuss in Forum

15. What will be the output of the program ?

```
#include<stdio.h>

int main()
{
    union var
    {
        int a, b;
    };
    union var v;
    v.a=10;
    v.b=20;
    printf("%d\n", v.a);
    return 0;
}
```

- [ ] A.10 ✗
- [✓] B.20 ✓
- [ ] C.30 ✗
- [ ] D.0 ✗

Your Answer: Option B

Correct Answer: Option B

Learn more problems on : Structures, Unions, Enums

Discuss about this problem : Discuss in Forum

16. Nested unions are allowed

- [✓] A.True ✓
- [ ] B.False ✗

Your Answer: Option A

Correct Answer: Option A

Learn more problems on : Structures, Unions, Enums

Discuss about this problem : Discuss in Forum

17. Can we have an array of bit fields?

- [ ] A.Yes ✗
- [✓] B.No ✓

Your Answer: Option B

Correct Answer: Option B

Learn more problems on : Structures, Unions, Enums

Discuss about this problem : Discuss in Forum

18. To scan *a* and *b* given below, which of the following *scanf()* statement will you use?

```
#include<stdio.h>

float a;
double b;
```

- [ ] A.scanf("%f %f", &a, &b); ✗
- [ ] B.scanf("%Lf %Lf", &a, &b); ✗
- [✓] C.scanf("%f %Lf", &a, &b); ✗
- [ ] D.scanf("%f %lf", &a, &b); ✓

Your Answer: Option C

Correct Answer: Option D

Explanation:

To scan a float value, *%f* is used as format specifier.

To scan a double value, *%lf* is used as format specifier.

Therefore, the answer is *scanf("%f %lf", &a, &b);*

Learn more problems on : Input / Output

Discuss about this problem : Discuss in Forum

19. What will be the output of the program?

```
#define P printf("%d\n", -1^~0);
#define M(P) int main()\
             {\
                P\
                return 0;\
             }
M(P)
```

- ☐ A. 1 ✗
- ☑ B. 0 ✓
- ☐ C. -1 ✗
- ☐ D. 2 ✗

Your Answer: Option B

Correct Answer: Option B

Learn more problems on : Bitwise Operators

Discuss about this problem : Discuss in Forum

20. What will be the output of the program?

```
#include<stdio.h>

int main()
{
    unsigned int res;
    res = (64 >>(2+1-2)) & (~(1<<2));
    printf("%d\n", res);
    return 0;
}
```

- ☑ A. 32 ✓
- ☐ B. 64 ✗
- ☐ C. 0 ✗
- ☐ D. 128 ✗

Your Answer: Option A

Correct Answer: Option A

Learn more problems on : Bitwise Operators

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