Online C Programming Test :: C Programming Test - Random

Marks : 13/20		
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.What is the output of the program?

```
#include<stdio.h>
int main()
{
    union a
    {
        int i;
        char ch[2];
    };
    union a u;
    u.ch[0] = 3;
    u.ch[1] = 2;
    printf("%d, %d, %d\n", u.ch[0], u.ch[1], u.i);
    return 0;
}
```

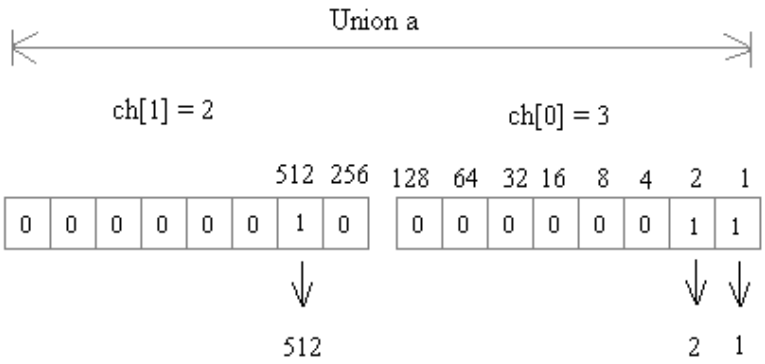
- [x] A.3, 2, 515
- [ ] B.515, 2, 3
- [ ] C.3, 2, 5
- [ ] D.None of these

Your Answer: Option A

Correct Answer: Option A

Explanation:

printf("%d, %d, %d\n", u.ch[0], u.ch[1], u.i); It prints the value of *u.ch[0] = 3, u.ch[1] = 2* and it prints the value of *u.i* means the value of entire union size.



So, 512 + 2 + 1 = 515

i = 515

So the output is 3, 2, 515.

Learn more problems on : Declarations and Initializations

Discuss about this problem : Discuss in Forum

2. Bitwise can be used to generate a random number.
   - 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

3. What will be the output of the program ?

```
#include<stdio.h>

int main()
{
    int i=32, j=0x20, k, l, m;
    k=i|j;
    l=i&j;
    m=k^l;
    printf("%d, %d, %d, %d, %d\n", i, j, k, l, m);
    return 0;
}
```

   - A. 0, 0, 0, 0, 0 ✗
   - B. 0, 32, 32, 32, 32 ✗
   - C. 32, 32, 32, 32, 0 ✓
   - D. 32, 32, 32, 32, 32 ✗

   Your Answer: Option C

   Correct Answer: Option C

Learn more problems on : Bitwise Operators

Discuss about this problem : Discuss in Forum

4. Assunming, integer is 2 byte, What will be the output of the program?

```
#include<stdio.h>

int main()
{
    printf("%x\n", -1>>1);
    return 0;
}
```

- [x] A. ffff
- [ ] B. 0fff
- [ ] C. 0000
- [ ] D. fff0

Your Answer: Option A

Correct Answer: Option A

Explanation:

Negative numbers are treated with 2's complement method.

1's complement: Inverting the bits (all 1s to 0s and all 0s to 1s)
2's complement: Adding 1 to the result of 1's complement.

```
Binary of 1(2byte)     :  0000 0000 0000 0001
Representing -1:
1s complement of 1(2byte)    : 1111 1111 1111 1110
Adding 1 to 1's comp. result : 1111 1111 1111 1111
Right shift 1bit(-1>>1): 1111 1111 1111 1111 (carry out 1)
Hexadecimal            : f   f    f    f
(Filled with 1s in the left side in the above step)
```

Note:

1. Fill with 1s in the left side for right shift for negative numbers.
2. Fill with 0s in the right side for left shift for negative numbers.
3. Fill with 0s in the left side for right shift for positive numbers.
4. Fill with 0s in the right side for left shift for positive numbers.

Learn more problems on : Bitwise Operators

Discuss about this problem : Discuss in Forum

5. What will be the output of the program ?

```
#include<stdio.h>

int main()
{
    printf("India", "BIX\n");
    return 0;
}
```

- [ ] A. Error
- [x] B. India BIX
- [ ] C. India

D.BIX ✖

Your Answer: Option B

Correct Answer: Option C

Explanation:

printf("India", "BIX\n"); It prints "India". Because ,(comma) operator has Left to Right associativity. After printing "India", the statement got terminated.

Learn more problems on : Strings

Discuss about this problem : Discuss in Forum

6. What will be the output of the program ?

```
#include<stdio.h>

int main()
{
    char t;
    char *p1 = "India", *p2;
    p2=p1;
    p1 = "BIX";
    printf("%s %s\n", p1, p2);
    return 0;
}
```

A.India BIX ✖
B.BIX India ✔
C.India India ✖
D.BIX BIX ✖

Your Answer: Option D

Correct Answer: Option B

Explanation:

Step 1: *char *p1 = "India", *p2;* The variable *p1* and *p2* is declared as an pointer to a character value and *p1* is assigned with a value "India".

Step 2: *p2=p1;* The value of *p1* is assigned to variable *p2*. So *p2* contains "India".

Step 3: *p1 = "BIX";* The *p1* is assigned with a string "BIX"

Step 4: *printf("%s %s\n", p1, p2);* It prints the value of *p1* and *p2*.

Hence the output of the program is "BIX India".

Learn more problems on : Strings

Discuss about this problem : Discuss in Forum

7. *typedef*'s have the advantage that they obey scope rules, that is they can be declared local to a function or a block whereas *#define*'s always have a global effect.

A.Yes ✔
B.No ✖

Your Answer: Option A

Correct Answer: Option A

Learn more problems on : Typedef

Discuss about this problem : Discuss in Forum

8. A *float* is 4 bytes wide, whereas a *double* is 8 bytes wide.
   - A. True ✓
   - B. False ✗

   Your Answer: Option A

   Correct Answer: Option A

   Explanation:

   True,
   float = 4 bytes.
   double = 8 bytes.

   Learn more problems on : Declarations and Initializations

   Discuss about this problem : Discuss in Forum

9. What will be the output of the program If the integer is 4bytes long?

```
#include<stdio.h>

int main()
{
    int ***r, **q, *p, i=8;
    p = &i;
    q = &p;
    r = &q;
    printf("%d, %d, %d\n", *p, **q, ***r);
    return 0;
}
```

   - A. 8, 8, 8 ✓
   - B. 4000, 4002, 4004 ✗
   - C. 4000, 4004, 4008 ✗
   - D. 4000, 4008, 4016 ✗

   Your Answer: Option A

   Correct Answer: Option A

   Learn more problems on : Pointers

   Discuss about this problem : Discuss in Forum

10. How many bytes of memory will the following code reserve?

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

int main()
```

```
{
    int *p;
    p = (int *)malloc(256 * 256);
    if(p == NULL)
        printf("Allocation failed");
    return 0;
}
```

- A. 65536 ✗
- B. Allocation failed ✓
- C. Error ✗
- D. No output ✗

Your Answer: Option D

Correct Answer: Option B

Explanation:

Hence 256*256 = 65536 is passed to *malloc()* function which can allocate upto 65535. So the memory allocation will be failed in 16 bit platform (Turbo C in DOS).

If you compile the same program in 32 bit platform like Linux (GCC Compiler) it may allocate the required memory.

Learn more problems on : Memory Allocation

Discuss about this problem : Discuss in Forum

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

```
#include<stdio.h>
#include<stdlib.h>
#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;
}
```

- A. 2, 8 ✓
- B. 4, 16 ✗
- C. 8, 24 ✗
- D. 16, 32 ✗

Your Answer: Option B

Correct Answer: Option A

Learn more problems on : Memory Allocation

Discuss about this problem : Discuss in Forum

12. What will happen if in a C program you assign a value to an array element whose subscript exceeds the size of array?
    - A. The element will be set to 0. ✗
    - B. The compiler would report an error. ✗

C. The program may crash if some important data gets overwritten.
D. The array size would appropriately grow.

Your Answer: Option C

Correct Answer: Option C

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.

Learn more problems on : Arrays

Discuss about this problem : Discuss in Forum

13. 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

14. What will be the output of the program (myprog.c) given below if it is executed from the command line?
cmd> *myprog one two three*

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

```
int main(int argc, char *argv[])
{
    int i;
    for(i=1; i<argc; i++)
        printf("%c", argv[i][0]);
    return 0;
}
```

- A.oot ✗
- B.ott ✓
- C.nwh ✗
- D.eoe ✗

Your Answer: Option B

Correct Answer: Option B

Learn more problems on : Command Line Arguments

Discuss about this problem : Discuss in Forum

15. What will be the output of the program?

```
#include<stdio.h>

int main()
{
    const char *s = "";
    char str[] = "Hello";
    s = str;
    while(*s)
        printf("%c", *s++);

    return 0;
}
```

- A.Error ✗
- B.H ✗
- C.Hello ✓
- D.Hel ✗

Your Answer: Option A

Correct Answer: Option C

Explanation:

Step 1: *const char *s = "";* The constant variable *s* is declared as an pointer to an array of characters type and initialized with an empty string.

Step 2: *char str[] = "Hello";* The variable *str* is declared as an array of charactrers type and initialized with a string "Hello".

Step 3: *s = str;* The value of the variable *str* is assigned to the variable *s*. Therefore *str* contains the text "Hello".

Step 4: *while(*s){ printf("%c", *s++); }* Here the while loop got executed untill the value of the variable *s* is available and it prints the each character of the variable *s*.

Hence the output of the program is "Hello".

Learn more problems on : Const

Discuss about this problem : Discuss in Forum

16. If a function contains two *return* statements successively, the compiler will generate warnings. Yes/No ?

A. Yes ✅
B. No ✖

Your Answer: Option B

Correct Answer: Option A

Explanation:

Yes. If a function contains two *return* statements successively, the compiler will generate "Unreachable code" warnings.

Example:

```
#include<stdio.h>
int mul(int, int); /* Function prototype */

int main()
{
    int a = 4, b = 3, c;
    c = mul(a, b);
    printf("c = %d\n", c);
    return 0;
}
int mul(int a, int b)
{
   return (a * b);
   return (a - b); /* Warning: Unreachable code */
}
```

Output:
c = 12

Learn more problems on : Functions

Discuss about this problem : Discuss in Forum

17. What will be the output of the program?

```
#include<stdio.h>
#include<stdarg.h>
void dumplist(int, ...);

int main()
{
    dumplist(2, 4, 8);
    dumplist(3, 6, 9, 7);
    return 0;
}
void dumplist(int n, ...)
{
    va_list p; int i;
    va_start(p, n);

    while(n-->0)
    {
        i = va_arg(p, int);
        printf("%d", i);
```

```
    }
    va_end(p);
    printf("\n");
}
```

- A. 2 4
3 6 ✗
- B. 2 4 8
3, 6, 9, 7 ✗
- C. 4 8
6 9 7 ✓
- D. 1 1 1
1 1 1 1 ✗

Your Answer: Option B

Correct Answer: Option C

Learn more problems on : Variable Number of Arguments

Discuss about this problem : Discuss in Forum

18. In a call to *printf()* function the format specifier %b can be used to print binary equivalent of an integer.
    - A. True ✗
    - B. False ✓

    Your Answer: Option B

    Correct Answer: Option B

    Explanation:

    There is no format specifier named *%b* in c.

    Learn more problems on : Input / Output

    Discuss about this problem : Discuss in Forum

19. Point out the error in the program?

```
typedef struct data mystruct;
struct data
{
    int x;
    mystruct *b;
};
```

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

    Your Answer: Option C

    Correct Answer: Option C

    Explanation:

    Here the type name *mystruct* is known at the point of declaring the structure, as it is already defined.

Learn more problems on : Structures, Unions, Enums

Discuss about this problem : Discuss in Forum

20. 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

***** 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