Online C Programming Test :: C Programming Test 9

Marks : 15/20

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 will be the output of the program?

```
#include<stdio.h>
int main()
{
    int x=1, y=1;
    for(; y; printf("%d %d\n", x, y))
    {
        y = x++ <= 5;
    }
    printf("\n");
    return 0;
}
```

A.
2 1
3 1
4 1
5 1
6 1
7 0

B.
2 1
3 1
4 1
5 1
6 1

C.
2 1
3 1
4 1
5 1

D.
2 2
3 3
4 4
5 5

Your Answer: Option A

Correct Answer: Option A

Learn more problems on : Control Instructions

Discuss about this problem : Discuss in Forum

2. Which of the following is the correct order of evaluation for the below expression?

*z = x + y * z / 4 % 2 - 1*

- [x] A. * / % + - = ✓
- [ ] B. = * / % + - ✗
- [ ] C. / * % - + = ✗
- [ ] D. * % / - + = ✗

Your Answer: Option A

Correct Answer: Option A

Explanation:

C uses left associativity for evaluating expressions to break a tie between two operators having same precedence.

Learn more problems on : Expressions

Discuss about this problem : Discuss in Forum

3. Which of the following range is a valid *long double* (Turbo C in 16 bit DOS OS) ?

- [ ] A. $3.4E^{-4932}$ to $1.1E^{+4932}$ ✓
- [x] B. $3.4E^{-4932}$ to $3.4E^{+4932}$ ✗
- [ ] C. $1.1E^{-4932}$ to $1.1E^{+4932}$ ✗
- [ ] D. $1.7E^{-4932}$ to $1.7E^{+4932}$ ✗

Your Answer: Option B

Correct Answer: Option A

Explanation:

The range of *long double* is $3.4E^{-4932}$ to $1.1E^{+4932}$

Learn more problems on : Floating Point Issues

Discuss about this problem : Discuss in Forum

4. Will the *printf()* statement print the same values for any values of *a*?

```
#include<stdio.h>
int main()
{
    float a;
    scanf("%f", &a);
    printf("%f\n", a+a+a);
    printf("%f\n", 3*a);
    return 0;
}
```

✓

A. Yes
B. No ✖

Your Answer: Option B

Correct Answer: Option A

Learn more problems on : Floating Point Issues

Discuss about this problem : Discuss in Forum

5. What will be the output of the program?

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

int main()
{
    int i=0;
    i++;
    if(i<=5)
    {
        printf("IndiaBIX");
        exit(1);
        main();
    }
    return 0;
}
```

A. Prints "IndiaBIX" 5 times ✖
B. Function *main()* doesn't calls itself ✖
C. Infinite loop ✖
D. Prints "IndiaBIx" ✔

Your Answer: Option D

Correct Answer: Option D

Explanation:

Step 1: *int i=0;* The variable *i* is declared as in integer type and initialized to '0'(zero).

Step 2: *i++;* Here variable *i* is increemented by 1. Hence *i* becomes '1'(one).

Step 3: *if(i<=5)* becomes *if(1 <=5)*. Hence the *if* condition is satisfied and it enter into *if* block statements.

Step 4: *printf("IndiaBIX");* It prints "IndiaBIX".

Step 5: *exit(1);* This *exit* statement terminates the program execution.

Hence the output is "IndiaBIx".

Learn more problems on : Functions

Discuss about this problem : Discuss in Forum

6. What will be the output of the program ?

```
#include<stdio.h>

int main()
```

```
{
    int i, a[] = {2, 4, 6, 8, 10};
    change(a, 5);
    for(i=0; i<=4; i++)
        printf("%d, ", a[i]);
    return 0;
}
void change(int *b, int n)
{
    int i;
    for(i=0; i<n; i++)
        *(b+1) = *(b+i)+5;
}
```

- A.7, 9, 11, 13, 15 ✗
- B.2, 15, 6, 8, 10 ✓
- C.2 4 6 8 10 ✗
- D.3, 1, -1, -3, -5 ✗

Your Answer: Option B

Correct Answer: Option B

Learn more problems on : Pointers

Discuss about this problem : Discuss in Forum

7. 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;
}
```

- A.1200, 1202, 1204 ✗
- B.1200, 1200, 1200 ✓
- C.1200, 1204, 1208 ✗
- D.1200, 1202, 1200 ✗

Your Answer: Option D

Correct Answer: Option B

Explanation:

Step 1: *int arr[]={2, 3, 4, 1, 6};* The variable *arr* is declared as an integer array and initialized.

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

Learn more problems on : Arrays

Discuss about this problem : Discuss in Forum

8. Which of the following is correct way to define the function *fun()* in the below program?

```
#include<stdio.h>

int main()
{
    int a[3][4];
    fun(a);
    return 0;
}
```

- [x] A.
```
void fun(int p[][4])
{
}
```
- [ ] B.
```
void fun(int *p[4])
{
}
```
- [ ] C.
```
void fun(int *p[][4])
{
}
```
- [ ] D.
```
void fun(int *p[3][4])
{
}
```

Your Answer: Option A

Correct Answer: Option A

Explanation:

void fun(int p[][4]){ } is the correct way to write the function *fun()*. while the others are considered only the function *fun()* is called by using *call by reference*.

Learn more problems on : Arrays

Discuss about this problem : Discuss in Forum

9. What will be the output of the program ?

```
#include<stdio.h>

int main()
{
    printf(5+"IndiaBIX\n");
    return 0;
}
```

- [ ] A. Error
- [ ] B. IndiaBIX
- [x] C. BIX
- [ ] D. None of above

Your Answer: Option C

Correct Answer: Option C

Explanation:

printf(5+"IndiaBIX\n"); In the printf statement, it skips the first 5 characters and it prints "BIX"

Learn more problems on : Strings

Discuss about this problem : Discuss in Forum

10. What will be the output of the program ?

```
#include<stdio.h>

int main()
{
    enum status {pass, fail, absent};
    enum status stud1, stud2, stud3;
    stud1 = pass;
    stud2 = absent;
    stud3 = fail;
    printf("%d %d %d\n", stud1, stud2, stud3);
    return 0;
}
```

- A. 0, 1, 2 ✗
- B. 1, 2, 3 ✗
- C. 0, 2, 1 ✓
- D. 1, 3, 2 ✗

Your Answer: Option C

Correct Answer: Option C

Learn more problems on : Structures, Unions, Enums

Discuss about this problem : Discuss in Forum

11. 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 A

Correct Answer: Option A

Explanation:

The structure *emp* contains a member *e* of the same type.(i.e) *struct emp*. At this stage compiler does not know the size of sttructure.

Learn more problems on : Structures, Unions, Enums

Discuss about this problem : Discuss in Forum

12. What will be the output of the program ?

```
#include<stdio.h>

int main()
```

```
{
    FILE *ptr;
    char i;
    ptr = fopen("myfile.c", "r");
    while((i=fgetc(ptr))!=NULL)
        printf("%c", i);
    return 0;
}
```

- A. Print the contents of file "myfile.c" ✗
- B. Print the contents of file "myfile.c" upto NULL character ✗
- C. Infinite loop ✓
- D. Error in program ✗

Your Answer: Option C

Correct Answer: Option C

Explanation:

The program will generate infinite loop. When an EOF is encountered *fgetc()* returns EOF. Instead of checking the condition for EOF we have checked it for NULL. so the program will generate infinite loop.

Learn more problems on : Input / Output

Discuss about this problem : Discuss in Forum

13. While calling the *fprintf()* function in the format string conversion specifier *%s* can be used to write a character string in capital letters.
- A. True ✗
- B. False ✓

Your Answer: Option A

Correct Answer: Option B

Explanation:

The *%s* format specifier tells the compiler the given input was string of characters.

Learn more problems on : Input / Output

Discuss about this problem : Discuss in Forum

14. The maximum combined length of the command-line arguments including the spaces between adjacent arguments is
- A. 128 characters ✗
- B. 256 characters ✗
- C. 67 characters ✗
- D. It may vary from one operating system to another ✓

Your Answer: Option D

Correct Answer: Option D

Learn more problems on : Command Line Arguments

Discuss about this problem : Discuss in Forum

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

```
#include<stdio.h>
char *fun(unsigned int num, int base);

int main()
{
    char *s;
    s=fun(128, 2);
    s=fun(128, 16);
    printf("%s\n",s);
    return 0;
}
char *fun(unsigned int num, int base)
{
    static char buff[33];
    char *ptr = &buff[sizeof(buff)-1];
    *ptr = '\0';
    do
    {
        *--ptr = "0123456789abcdef"[num %base];
        num /=base;
    }while(num!=0);
    return ptr;
}
```

- [x] A. It converts a number to a given base.
- [ ] B. It converts a number to its equivalent binary.
- [ ] C. It converts a number to its equivalent hexadecimal.
- [ ] D. It converts a number to its equivalent octal.

Your Answer: Option A

Correct Answer: Option A

Learn more problems on : Bitwise Operators

Discuss about this problem : Discuss in Forum

16. What will be the output of the program?

```
#include<stdio.h>

int main()
{
    const c = -11;
    const int d = 34;
    printf("%d, %d\n", c, d);
    return 0;
}
```

- [ ] A. Error
- [x] B. -11, 34
- [ ] C. 11, 34
- [ ] D. None of these

Your Answer: Option B

Correct Answer: Option B

Explanation:

Step 1: *const c = -11;* The constant variable *'c'* is declared and initialized to value "-11".

Step 2: *const int d = 34;* The constant variable *'d'* is declared as an integer and initialized to value '34'.

Step 3: *printf("%d, %d\n", c, d);* The value of the variable *'c'* and *'d'* are printed.

Hence the output of the program is -11, 34

Learn more problems on : Const

Discuss about this problem : Discuss in Forum

17. Point out the correct statement will let you access the elements of the array using 'p' in the following program?

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

int main()
{
    int i, j;
    int(*p)[3];
    p = (int(*)[3])malloc(3*sizeof(*p));
    return 0;
}
```

A.
```
for(i=0; i<3; i++)
{
    for(j=0; j<3; j++)
        printf("%d", p[i+j]);
}
```
✗

B.
```
for(i=0; i<3; i++)
    printf("%d", p[i]);
for(i=0; i<3; i++)
{
    for(j=0; j<3; j++)
        printf("%d", p[i][j]);
}
```
✗

C.
```
for(j=0; j<3; j++)
    printf("%d", p[i][j]);
```
✓

D.
```
for(j=0; j<3; j++)
    printf("%d", p[i][j]);
```
✗

Your Answer: Option C

Correct Answer: Option C

Learn more problems on : Memory Allocation

Discuss about this problem : Discuss in Forum

18. What do the following declaration signify?

```
char *arr[10];
```

A. *arr* is a array of 10 character pointers. ✓
B. *arr* is a array of function pointer. ✗
C. *arr* is a array of characters. ✗
D. *arr* is a pointer to array of characters. ✗

Your Answer: Option D

Correct Answer: Option A

Learn more problems on : Complicated Declarations

Discuss about this problem : Discuss in Forum

19. Point out the error in the following program.

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

int main()
{
    char str1[] = "Learn through IndiaBIX\0.com",  str2[120];
    char *p;
    p = (char*) memccpy(str2, str1, 'i', strlen(str1));
    *p = '\0';
    printf("%s", str2);
    return 0;
}
```

- A. Error: in *memccpy* statement ✖
- B. Error: invalid pointer conversion ✖
- C. Error: invalid variable declaration ✖
- D. No error and prints "Learn through Indi" ✔

Your Answer: Option D

Correct Answer: Option D

Explanation:

Declaration:
*void *memccpy(void *dest, const void *src, int c, size_t n);* : Copies a block of n bytes from src to dest

With *memccpy()*, the copying stops as soon as either of the following occurs:
=> the character 'i' is first copied into *str2*
=> n bytes have been copied into *str2*

Learn more problems on : Library Functions

Discuss about this problem : Discuss in Forum

20. Point out the error in the following program.

```
#include<stdio.h>

int main()
{
    char str[] = "IndiaBIX";
    printf("%.#s %2s", str, str);
    return 0;
}
```

- A. Error: in Array declaration ✖
- B. Error: *printf* statement ✖
- C. Error: unspecified character in *printf* ✖
- D. No error ✔

Your Answer: Option D

Correct Answer: Option D

Learn more problems on : Library Functions

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