

Học phần: Lập trình cơ bản
Thời gian: 45 phút

Họ tên:

Ngày sinh:

Lớp:

Ngày thi:

Hãy khoanh tròn vào đáp án đúng

1. Máy tính điện tử có thể cho phép

- a) Xử lý thông tin b) Lưu trữ thông tin c) Soạn thảo văn bản d) Tất cả các phương án trên

2. Điều không phải là tính chất của giải thuật

- a) Xác định b) Khả thi c) Phổ dụng d) Số lượng các bước có thể vô hạn

3. Virus máy tính là

- a) Một chương trình b) Một thiết bị ngoại vi c) Một virus sinh học d) Một chất ăn mòn kim loại

4. Đây là cách đặt tên sai trong ngôn ngữ lập trình C

- a) _abc b) x1y c) d#k d) u2_v

5. Tìm giá trị của 25_{10} trong hệ cơ số 2

- a) 0001 b) 11001 c) 10011 d) 1110

6. Tìm giá trị của 34_{10} trong hệ cơ số 16

- a) 22 b) 12 c) 120 d) 202

7. Xác định kết quả của thuật toán sau

Bước 1. Cho $N = 10$;

Bước 2. Cho $i = 1$;

Bước 3. Nếu $i > N$ thì chuyển sang bước 5, ngược lại chuyển sang bước 4;

Bước 4. Cho $i = i + 2$. Quay lại bước 3;

Bước 5. In ra i , kết thúc thuật toán.

- a) 9 b) 11 c) 13 d) 10

8. Hãy xác định giá trị của a và b theo thuật toán sau:

Bước 1. Cho $a = 6$, $b = 8$;

Bước 2. Gán $c = a$, $d = b$;

Bước 3. Nếu $c = d$ thì chuyển sang bước 6, ngược lại chuyển sang bước 4;

Bước 4. Nếu $c > d$ thì gán $c = c - d$, quay lại bước 3, ngược lại chuyển sang bước 5;

Bước 5. Gán $d = d - c$, quay lại bước 3;

Bước 6. Gán $a = a/c$; $b = b/d$. In ra a và b, kết thúc thuật toán.

- a) $a = 2$, $b = 4$. b) $a = 1$, $b = 2$. c) $a = 3$, $b = 4$ d) $a = 2$, $b = 3$.

9. Cách khai báo đúng cho biến sum có kiểu dữ liệu nguyên trong ngôn ngữ lập trình C

- a) sum : integer; b) integer sum; c) int sum; d) sum int;

10. Cách in đúng một biến kiểu nguyên *sum* ra màn hình

a) `printf("%s", sum);` b) `print("%i", sum);` c) `printf("%d", sum);` d) `printf("%d", &sum);`

11. Hãy xác định mục tiêu của đoạn mã sau:

```
if( (flag == 1) || (letter != 'X') )
    exit_flag = 0;
else
    exit_flag = 1;
```

- a) Nếu *flag* bằng 1 hoặc *letter* không phải là 'X' thì gán giá trị 0 cho *exit_flag*, trong trường hợp ngược lại thiết lập *exit_flag* bằng 1.
- b) Nếu *flag* bằng 1 hoặc *letter* không phải là 'X' thì gán giá trị 1 cho *exit_flag*, trong trường hợp ngược lại thiết lập *exit_flag* bằng 0.
- c) Nếu *flag* bằng 1 và *letter* không phải là 'X' thì gán giá trị 0 cho *exit_flag*, trong trường hợp ngược lại thiết lập *exit_flag* bằng 1.
- d) Nếu *flag* không bằng 1 và *letter* không phải là 'X' thì gán giá trị 0 cho *exit_flag*, trong trường hợp ngược lại thiết lập *exit_flag* bằng 1.

12. Hãy xác định kết quả của đoạn mã sau:

```
for( count = 1; count <= 10; count = count + 1)
    printf("%d\n", count);
```

- a) In các giá trị từ 1 đến 10, mỗi giá trị trên một dòng
- b) In các giá trị từ 1 đến 10 trên một dòng, các giá trị ngăn cách với nhau bởi 1 dấu cách
- c) In các giá trị từ 1 đến 10 trên một dòng
- d) In các giá trị từ 1 đến 11, mỗi giá trị trên một dòng

13. Hãy xác định kết quả của đoạn mã sau:

```
count = 1;
while( count <= 10 ) {
    printf("%d", count);
    count = count + 1;
}
```

- a) In các giá trị từ 1 đến 10 ra màn hình dưới dạng như sau: 12345678910
- b) In các giá trị từ 1 đến 10 ra màn hình, mỗi giá trị trên một dòng
- c) In các giá trị từ 1 đến 11 ra màn hình, mỗi giá trị trên một dòng
- d) In các giá trị từ 1 đến 11 ra màn hình dưới dạng như sau: 1234567891011

14. Hãy xác định kết quả in ra màn hình của đoạn mã sau:

```
a = 1;
while( a <= 5 ) {
    b = 1;
    while( b <= a ) {
        printf("%d", a);
        b = b + 1;
    }
    printf("\n");
    a = a + 1;
}
```

a)

```
1
22
333
4444
55555
```

b)

```
11
22
33
44
55
```

c)

```
11111
2222
333
44
5
```

d) Lỗi biên dịch

15. Hãy xác định kết quả in ra màn hình của đoạn mã sau

```
#include<stdio.h>
int main(){
    int arr[3]={ 10,20,30};
    int x=0;
    x = arr[++x];
    printf("%d",x);
    return 0;
}
```

a) 22 b) 20 c) 43 d) 44

16. Kết quả thực hiện của đoạn mã sau:

```
#include<stdio.h>
int main(){
    int arr[3][2]={ 10,20,30,40,50,60};
    int x=0;
    x = arr[1][0]*arr[0][1]+arr[2][1];
    printf("%d ",x);
    return 0;
}
```

- a) 660
- b) 2000
- c) Lỗi khi chạy
- d) Lỗi biên dịch

17. Cách gán chuỗi ký tự "Welcome" cho mảng ký tự *stuff* (không phải tại thời điểm khai báo)

- a) strcpy(stuff, 'Welcome');
- b) stuff = "Welcome";
- c) stuff[0] = "Welcome";
- d) strcpy(stuff, "Welcome");

18. Kết quả thực hiện của đoạn mã sau

```
int count = 10, *temp, sum = 0;
temp = &count;
*temp = 20;
temp = &sum;
*temp = count;
printf("count = %d, *temp = %d, sum = %d\n", count, *temp, sum );
```

- a) count = 2, *temp = 10, sum = 10
- b) count = 20, *temp = 20, sum = 20
- c) count = 10, *temp = 2, sum = 10
- d) count = 200, *temp = 0.2, sum = 1

19. Kết quả của việc thực hiện đoạn mã sau

```
#include<stdio.h>
int kk = 6, ll = 7;
int addmult(int ii, int jj)
{
    int kk, ll;
    kk = ii + jj;
    ll = ii * jj;
    return 0;
}
int main()
{
    int i=3, j=4, k, l;
    addmult(i, j);
    addmult(i, j);
    printf("%d, %d\n", kk, ll);
    return 0;
}
```

- a) 12, 12
- b) 7, 7
- c) 12, 7
- d) 6, 7

20. Chương trình sau thực hiện công việc gì:

```
#include<stdio.h>

int main()
{
    FILE *fp;
    char ch;
    int i=0;
    fp = fopen("myfile.c", "r");
    while((ch=getc(fp))!=EOF)
    {
        if(ch == '\n')
            i++;
    }
    fclose(fp);
    printf("%d", i);
    return 0;
}
```

- a) Đoạn mã tính số ký tự xuống dòng trong file myfile.c
- b) Đoạn mã tính số từ trong file myfile.c
- c) Đoạn mã tính số dòng trắng (dòng chỉ chứa các ký tự rỗng) trong file myfile.c
- d) Đoạn mã tính số dòng trong file myfile.c

Thí sinh được sử dụng tài liệu

-Hết-