

NỘI DUNG ÔN TẬP CUỐI KỲ

MÔN PHƯƠNG PHÁP LẬP TRÌNH

HỌC KỲ 1 – NĂM HỌC: 2022 - 2023

Nội dung ôn thi Cuối Kỳ môn Phương pháp lập trình

1. Các khái niệm lý thuyết cơ bản
2. Hàm (chương trình con)
3. Câu lệnh **if, if-else, switch-case**.
4. Câu lệnh **for, while, do while**
5. Câu lệnh **break, continue**
6. Con trỏ, Đệ quy
7. Sắp xếp và tìm kiếm
8. Mảng một chiều, nhiều chiều
9. Ký tự và chuỗi ký tự
10. Structures

Bài ôn tập:

A. Các khái niệm lý thuyết cơ bản

1. Những tên biến nào dưới đây được viết đúng theo quy tắc đặt tên của ngôn ngữ lập trình C.
i. mon hoc ii. 5monhoc iii. _monhoc iv. -monhoc
2. Điền vào kiểu dữ liệu và kích thước (sizeof) tương ứng với từng đặc tả trong bảng sau:

Đặc tả	Kiểu dữ liệu	Kích thước (sizeof)	Đặc tả	Kiểu dữ liệu	Kích thước (sizeof)
%d	int	4byte	%s	string	1byte
%ld	long	8byte	%p	pointer	
%f	float	4byte	%u	unsign	
%lf	double	8byte	%x	pointer	
%c	char	1byte			

3. Cho biết các phát biểu sau ĐÚNG hay SAI (không cần giải thích)
 1. Ngôn ngữ lập trình C được Dennis phát triển dựa trên ngôn ngữ lập trình B.
 2. Một biến được gọi là biến toàn cục nếu nó được khai báo ngoài tất cả các hàm kể cả hàm main().
 3. Một biến được gọi là một biến địa phương nếu nó được khai báo bên trong hàm main().
 4. Nếu hàm được gọi trước khi nó định nghĩa thì điều kiện là kiểu đầu vào của hàm phải là kiểu void.

5. Kiểu dữ liệu float có thể xử lý dữ liệu trong phạm vi -32768 đến 32767.
6. Kiểu dữ liệu mảng không được coi là kiểu dữ liệu cơ bản trong ngôn ngữ lập trình C.
7. Lệnh "continue" là lệnh cho phép chuyển sang vòng lặp tiếp theo mà không cần phải thực hiện phần còn lại của vòng lặp.
8. Câu lệnh `ch='A'`. Vậy `ch` sẽ chứa 8 byte.
9. Độ ưu tiên đối với các toán tử logic lần lượt là OR, AND, NOT.
10. Có 4 loại error trong lập trình: Run-time Error, Compile-time Error, Logical Error, Syntax Error.
11. Vòng lặp `do...while()` thực thi tối thiểu 2 lần.
12. Khi truyền tham số bằng tham chiếu, giá trị của biến sẽ chỉ thay đổi trong các hàm hiện hành.
13. Một biến kiểu **struct** làm tham số của hàm thì biến đó LUÔN truyền bằng trị.
14. Toán tử là các kí hiệu đặc trưng cho các phép toán trên dữ liệu.
15. Trong một số hệ thống máy tính, kích thước của biến con trỏ là 2 byte.
16. Cho `i` là số nguyên, phép toán `++i` và `i++` đều cho kết quả như nhau.
17. Kết quả của phép toán logic (`false || true`) là `false`.
18. Có 2 cách ép kiểu: ép kiểu tường minh, ép kiểu phi tường minh.
19. Cho `char* x = "DHTDT"`, kết quả của lệnh `printf("%c", *x)` là "DHTDT".
20. Cho mảng `int a[5]`, lệnh `sizeof(a)` sẽ trả về số phần tử của mảng.
21. Một hàm không thể được định nghĩa bên trong một hàm khác.
22. Hàm không thể trả về (return) nhiều hơn một giá trị tại một thời điểm.
23. Tất cả các hàm ngoại trừ hàm `main()` có thể được gọi đệ quy.
24. Tham số của hàm có thể được truyền bằng tham trị hoặc tham chiếu.
25. Kiểu dữ liệu `char` có độ rộng 8 bit.
26. Khai báo một mảng nhiều chiều có dạng
$$\text{int arr}[2][2][3] = \{ \{ \{ 1, 2, 3 \}, \{ 4, 5, 6 \} \}, \{ \{ 7, 8, 9 \}, \{ 10, 11, 12 \} \} \}.$$
27. Biểu thức `*ptr++` và `++*ptr` là giống nhau.
28. Câu lệnh `max = a > b ? a > c ? a : c : b > c ? b : c` có cú pháp đúng.
29. Phép toán gán là phép toán kết hợp phải (right-to-left).
30. Cấu trúc `switch...case` có thể không cần lệnh `default`.

B. Cho biết kết quả in ra màn hình của các chương trình sau là gì:

1. Kết quả in ra màn hình của chương trình sau là gì:

```
#include <stdio.h>
void main()
{
    int i;
    for(i='f';i<'i';i++)
    {
        printf("%d",i);
    }
}
```

f = 102
i = 105
-> 102103104

2. Kết quả in ra màn hình của chương trình sau là gì:

```
#include <stdio.h>
void main()
{
    int a = 20, b = 21;
    int *c = &a, *d = &b;
    (*d)++;
    *c = *c + *d;
    printf("a = %d, b = %d, c = %d, d = %d",a,b,*c,*d);
}
```

c=20
d=21
d=b=22
c=42=a
->a = 42, b = 22, c=42, d = 22

3. Kết quả in ra màn hình của chương trình sau là gì:

```
#include <stdio.h>
void main()
{
    int i, a, b;
    i = 2023;
    a = 2 + ++i;
    b = 2 + --i;
    printf("a = %d,b = %d",a,b);
}
```

a = 2026
b = 2025

4. Kết quả in ra màn hình của chương trình sau là gì:

<pre>#include<stdio.h> int f(int a, int b) { if(a > b) return 0; if(a == b) return 1; return b + f(a,b - 1); }</pre>	<pre>int main() { printf("%d\n",f(10,20)); printf("%d\n",f(30,40)); printf("%d", f(50,60)); return 0; }</pre>
---	---

5. Kết quả in ra màn hình của chương trình sau là gì:

```
#include <stdio.h>
void main()
{
    int n1 = 365, n2 = 850;
    while(n1 != n2)
    {
        if(n1 > n2)
            n1 -= n2;
        if(n2 > n1)
            n2 -= n1;
    }
    printf("n1 = %d,n2 = %d",n1,n2);
}
```

6. Kết quả in ra màn hình của chương trình sau là gì:

```
#include <stdio.h>
void main() {
    int a,b,c,d;
    a = b = 404;
    c = a + a++ + --a + ++a + a--;
    d = b + ++b + b-- + b++ + --b;
    printf("ket qua c= %d, d= %d \n",c,d);
    if(c == d){
        printf("ket qua c bang d);
    }
    else{
        printf("ket qua c khac d);
    }
}
```

error syntax

7. Kết quả in ra màn hình của chương trình sau là gì:

```
#include <stdio.h>
void main()
{
    int x, y, z, a = 4, b = -2 , c = 0;
    x = ((a < b || b > c) && a != b);
    y = ((a > b || b < c) && a == b);
    z = (b < (a > b)) || ((a > b) && !(b > c));
    printf("x = %d, y = %d, z = %d",x,y,z);
}
```

x = 0, y = 0, z = 1

8. Kết quả in ra màn hình của chương trình sau là gì:

```
#include <stdio.h>
#include <string.h>
void main()
{
    char s1[50] = " ", s2[50] = " ", s3 = "abcde";
    strcpy(s1,s3);
    strcpy(s3,s2);
    printf("s1: %s s2: %s s3: %s",s1,s2,s3);
}
```

s3 thiếu []

9. Kết quả in ra màn hình của chương trình sau là gì:

```
#include <stdio.h>
void main()
{
    int i,s = 0;
    for (i=1; i<=9; i++)
    {
        if(i%2 == 0)
            break;
        s = s + i;
    }
    printf("%d",s);
}
```

1

10. Kết quả in ra màn hình của chương trình sau là gì:

```
#include <stdio.h>
void main()
{
    int i,s = 0;
    for (i=2; i<=6; i+=2)
    {
        if(i%2 == 0)
            continue;
        s = s + i;
    }
    printf("%d",s);
}
```

0

C. Viết hàm sử dụng các câu lệnh: if; if - else; switch - case; continue; break; for; while; do – while.

Lưu ý: Cần tuân thủ nguyên tắc khi thiết kế hàm sau đây:

- Không được nhập giá trị tham số trong hàm, các giá trị tham số được truyền từ lời gọi hàm.
- Không được in kết quả trong hàm mà phải trả kết quả về nơi gọi hàm thông qua câu lệnh return hoặc truyền tham chiếu (pass-by-reference)

1. Viết hàm tính: (mỗi câu là một hàm)

a. $S = 1 * 2 * 3 * \dots * n \text{ (n!)}$

b. $S = 1^2 + 2^2 + 3^2 + \dots + n^2$

c. $S = \frac{1}{2} + \frac{2}{3} + \frac{3}{4} + \dots + \frac{n-1}{n}$

d. $S = 1 + 3 + 5 + \dots + (2 * n + 1)$

e. $S = \sqrt{2} + \sqrt{3} + \sqrt{4} + \dots + \sqrt{n+1}$

f. $S = \sum_{i=1}^n i$

g. $S = \sum_{i=1}^n (3 * i - \frac{1}{2})$

2. Viết hàm tìm ước chung lớn nhất theo thuật toán: (mỗi câu là 1 hàm)

a. Euclide (chia lấy dư)

b. Trừ

3. Viết hàm tìm bội chung nhỏ nhất.

4. Viết hàm tính tổng các số nhỏ hơn n và chia hết cho 3. (Ví dụ: n = **10** kết quả **18**)

5. Viết hàm tính tổng các chữ số trong số nguyên dương n

6. Viết hàm tính trung bình cộng của các chữ số trong số nguyên dương n.

7. Viết hàm kiểm tra số nguyên n có phải là số đối xứng hay không?

8. Viết hàm tìm chữ số lớn nhất trong số n.

9. Viết hàm kiểm tra số n có phải là số hoàn thiện hay không. Nếu n là số hoàn thiện thì trả về 1, không phải là số hoàn thiện trả về 0. Biết số hoàn thiện là số có tổng các ước số nhỏ hơn nó thì bằng nó. Ví dụ: 6 là số hoàn thiện vì 6 có các ước: {1, 2, 3, 6} mà tổng các ước nhỏ hơn 6 là $1 + 2 + 3 = 6$.

10. Viết hàm kiểm tra số nguyên dương n có phải là số toàn chẵn. Nếu n là số toàn chẵn thì trả về 1, không phải là số toàn chẵn trả về 0.

D. Viết hàm sử dụng Độ quy và kết hợp hàm main()

1. Viết hàm đệ quy theo công thức sau:

$$f(x) = \begin{cases} 1, & x = 0 \\ 2, & x = 1 \\ 2f(x-1) + 3f(x-2), & x \geq 1 \end{cases}$$

2. Viết hàm main() để nhập vào một số nguyên dương. Sau đó, thực hiện gọi hàm ở **câu 1** để tính và in kết quả ra màn hình.
3. Viết hàm đệ quy theo công thức sau:

$$f(x) = \begin{cases} 3, & x = 0 \\ 5, & x = 1 \\ f(x-1) + 2f(x-2), & x \geq 1 \end{cases}$$

4. Viết hàm main() để nhập vào một số nguyên dương. Sau đó, thực hiện gọi hàm ở **câu 3** để tính và in kết quả ra màn hình.
5. Viết hàm đệ quy Tính trung bình của mảng $a[5] = \{1, 2, 3, 4, 5\}$.
6. Viết hàm đệ quy Tìm số lớn nhất trong mảng $a[5] = \{1, 30, 20, 100, 96\}$.
7. Viết hàm đệ quy Tính giá trị biểu thức sau:

$$f(x) = \begin{cases} 1, & x = 0 \\ 2f(x-1) + \frac{3}{4}, & x > 0 \end{cases}$$

8. Viết hàm đệ quy kiểm tra một số nguyên n có phải là số đối xứng.
9. Viết hàm đệ quy Tính giá trị biểu thức:

$$P(n) = \sum_{i=1}^n \left(3i - \frac{1}{3} \right).$$

10. Cho mảng số nguyên có n phần tử, viết hàm đệ quy trả về tổng của các phần tử trong mảng.

E. Viết hàm thao tác trên mảng một chiều a có n phần tử.

1. Viết hàm tính tổng các phần tử có giá trị chẵn trong mảng a có n phần tử
2. Viết hàm tính đếm các phần tử có giá trị lẻ trong mảng a có n phần tử
3. Viết hàm đếm số phần tử bằng với giá trị X trong mảng a có n phần tử.
4. Viết hàm kiểm tra các giá trị trong mảng a có là bội số của 5 không. Hàm trả về 1 nếu các giá trị trong mảng a là bội số của 5, ngược lại trả về 0.

VD: a[5] = {5,10,15,20,25} -> Hàm trả về 1.

a[5] = {5,10,14,20,25} -> Hàm trả về 0.

5. Viết hàm tìm vị trí phần tử bằng với giá trị X trong mảng a có n phần tử. Hàm trả về -1 nếu không có, ngược lại hàm sẽ trả về vị trí tìm được.
6. Viết hàm xóa các phần tử bằng với giá trị X trong mảng a có n phần tử.
7. Viết hàm tìm số chẵn cuối cùng trong mảng a có n phần tử. Hàm trả về -1 nếu không có, ngược lại hàm sẽ trả về giá trị chẵn đầu tiên trong mảng.
8. Viết hàm tìm số lẻ lớn nhất trong mảng a có n phần tử. Hàm trả về 0 nếu không có, ngược lại hàm sẽ trả về số lẻ lớn nhất.
9. Viết hàm tính tổng các phần tử chia hết cho 3 và 5 trong mảng a có n phần tử.
10. Viết hàm đếm số lượng phần tử có giá trị trùng nhau trong mảng a có n phần tử.

F. Viết hàm thao tác trên chuỗi ký tự và sử dụng các hàm trong thư viện “string.h” (strcpy; strdup; strlwr;strupr; strcat; strrev; strstr; strlen; strcmp; strcmpi)

1. Viết hàm đếm số từ có trong chuỗi s. Ví dụ: s= “Lap trinh C” thì kết quả là 3
2. Viết hàm kiểm tra xem chuỗi s có đối xứng hay không
3. Viết hàm đếm số lượng ký tự là nguyên âm trong chuỗi s.
4. Viết hàm đếm ký tự viết hoa trong chuỗi s. Ví dụ: s= “Lap trinh C” kết quả 2.
5. Viết hàm viết hoa chữ cái đầu của các từ trong chuỗi s. Ví dụ: s= “khoa Cong nghe thong in” kết quả s= “Khoa Cong Nghe Thong Tin”.
6. Viết hàm nối chuỗi s2 vào trong chuỗi s1 ở 1 vị trí x. Ví dụ: s1= “Mon hoc qua de”; s2= “Lap trinh C ”; x = 4 kết quả chuỗi s1= “Mon Lap trinh C hoc qua de”
7. Viết hàm tách chuỗi s ra thành 3 chuỗi. Với s1 là họ, s2 là họ lót và s3 là tên trong chuỗi s. Ví dụ: s= “Nguyen Quoc Thuan”. s1= “Nguyen”; s2= “Quoc”; s3= “Thuan”.

8. Viết hàm kiểm tra chuỗi s có phải là chuỗi chứa “số căn cước công dân” hay không. Chuỗi căn cước công dân là chuỗi có chứa định dạng “XXX X XX XXXXXX” với X là một số từ 0 -> 9. Hàm trả về 1 nếu chuỗi đó có chứa “số căn cước công dân”, ngược lại hàm sẽ trả về 0.

Ví dụ: s= “so can cuoc la: 079 2 03 012345 cap tai HCM” -> 1

s= “can cuoc so: 064 3 03 123456 cap tai Gia Lai” -> 1

s= “CMND so: 026109897” -> 0

9. Viết hàm kiểm tra chuỗi s có phải là chuỗi chứa “địa chỉ” hay không. Chuỗi địa chỉ là chuỗi có chứa định dạng “Quan X” với X là một số từ 1->9. Hàm trả về 1 nếu chuỗi đó có chứa chuỗi địa chỉ, ngược lại hàm sẽ trả về 0.

Ví dụ: s= “dia chi: so 19 Nguyen Huu Tho, Quan 7, Tp.HCM” -> 1

s= “so 19 Nguyen Huu Tho, Quan 7, Thanh pho Ho Chi Minh” -> 1

s= “dia chi: so 7 Nguyen Huu Tho” -> 0

10. Viết hàm nhập vào một số nguyên dương n ($n < 1.000.000$) và một chuỗi s. Sau đó thực hiện nối số nguyên dương vừa nhập vào chuỗi s. Hàm trả về độ dài chuỗi s sau khi nối. Ví dụ: nhập n = 1234, chuỗi s = “ So vua nhap la: ” kết quả nhận được trong chuỗi s = “So vua nhap la: 1234”; kết quả hàm trả về 20.

G. STRUCTURES

1. Khai báo 1 cấu trúc **Sinh viên** gồm các thuộc tính sau

- Mã sinh viên (10 ký tự)
- Họ tên sinh viên (50 ký tự)
- Giới tính (1 ký tự)
- Năm sinh (số nguyên)
- Email (50 ký tự)
- Khoa (20 ký tự)

2. Viết hàm đếm số lượng sinh viên 20 tuổi (tính theo năm hiện tại là 2023) trong mảng một chiều a được khai báo theo kiểu dữ liệu **Sinh viên** (kiểu dữ liệu được định nghĩa ở câu 1) có n phần tử.

3. Viết hàm đếm số lượng sinh viên sử dụng Email có tên miền “@student.tdtu.edu.vn” trong mảng một chiều a được khai báo theo kiểu dữ liệu **Sinh viên** (kiểu dữ liệu được định nghĩa ở câu 1) có n phần tử.

4. Viết hàm đếm số lượng sinh viên đang học khoa Công nghệ thông tin trong mảng một chiều a được khai báo theo kiểu dữ liệu **Sinh viên** (kiểu dữ liệu được định nghĩa ở câu 1) có n phần tử.
5. Viết hàm sắp xếp danh sách sinh viên theo thứ tự năm sinh tăng dần trong mảng một chiều a được khai báo theo kiểu dữ liệu **Sinh viên** (kiểu dữ liệu được định nghĩa ở câu 1) có n phần tử.
6. Khai báo 1 cấu trúc **Độc giả** gồm các thuộc tính sau
 - Mã độc giả (7 ký tự)
 - Tên độc giả (20 ký tự)
 - Ngày mượn tài liệu (10 ký tự)
 - Số lượng tài liệu mượn (số nguyên)
7. Viết hàm đếm số lượng độc giả đang mượn trên 20 tài liệu trong mảng một chiều a được khai báo theo kiểu dữ liệu **Độc giả** (kiểu dữ liệu được định nghĩa ở câu 6) có n phần tử.
8. Viết hàm tính tổng số lượng tài liệu được mượn ngày “30/01/2023” trong mảng một chiều a được khai báo kiểu dữ liệu **Độc giả** (kiểu dữ liệu được định nghĩa ở câu 6) có n phần tử.
9. Viết hàm sắp xếp danh sách độc giả theo số lượng mượn tài liệu giảm dần trong mảng một chiều a được khai báo theo kiểu dữ liệu **Độc giả** (kiểu dữ liệu được định nghĩa ở câu 6) có n phần tử.
10. Viết hàm in 10 mã độc giả có số lượng tài liệu mượn nhiều nhất (biết rằng số lượng các tài liệu mượn này không trùng nhau) trong mảng một chiều a được khai báo theo kiểu dữ liệu **Độc giả** (kiểu dữ liệu được định nghĩa ở câu 6) có n phần tử.

Mọi thắc mắc liên hệ Thầy Nguyễn Quốc Thuận qua email: ngquochuan29@gmail.com

CHÚC CÁC BẠN THI
ĐẠT KẾT QUẢ TỐT