

Chương 9: Ôn tập

cuu duong than cong . com

Nhập môn lập trình

Trình bày: Nguyễn Sơn Hoàng Quốc

cuu duong than cong . com

Email: nshquoc@fit.hcmus.edu.vn

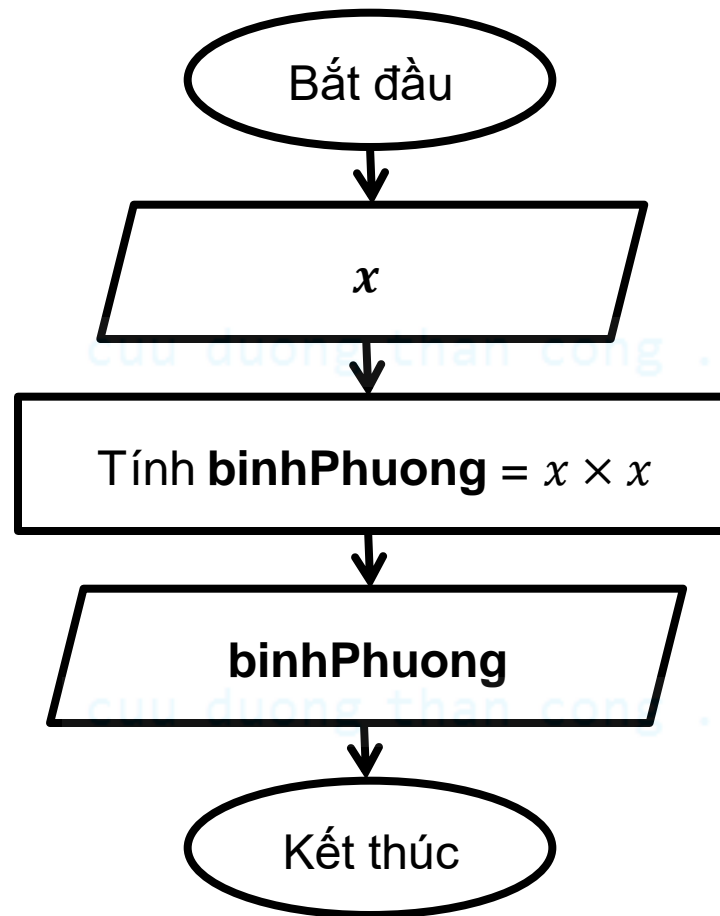
Bài 1: Vẽ lưu đồ và viết hàm bình phương

- Vẽ lưu đồ và viết hàm tính bình phương của một số x

cuu duong than cong . com

cuu duong than cong . com

Bài 1: Vẽ lưu đồ và viết hàm bình phương



Bài 1: Vẽ lưu đồ và viết hàm bình phương

```
1. int TinhBinhPhuong(int x)
2. {
3.     int binhPhuong;
4.
5.     binhPhuong = x * x;
6.
7.     return binhPhuong;
8. }
```

Câu 2 – Tính giai thừa

- Viết hàm tính giai thừa của một số nguyên dương n .

cuu duong than cong . com

cuu duong than cong . com

Câu 2 – Tính giai thừa

```
1. int TinhGiaiThua(int n)
2. {
3.     int giaiThua;
4.     cuu duong than cong . com
5.     giaiThua = 1;
6.     for (int i = 1; i <= n; i++)
7.         giaiThua = giaiThua * i;
8.     cuu duong than cong . com
9.     return giaiThua;
10. }
```

Câu 3 – Tổng bit

- Viết chương trình nhập vào một số nguyên dương x . Tính tổng các bit của số đó và xuất kết quả

cuu duong than cong . com

cuu duong than cong . com

Câu 3 – Tổng bit

1. #include <iostream>

2. using namespace std;

3. void Nhap(int &x);

4. int TinhTongBit(int x);

5. void Xuat(int);

Câu 3 – Tổng bit

```
1. void main()
2. {
3.     int x, tongBit;
4.
5.     cout << "Nhap x : ";
6.     Nhap(x);
7.
8.     tongBit = TinhTongBit(x);
9.
10.    cout << "Tong bit = ";
11.    Xuat(tongBit);
12. }
```

Câu 3 – Tổng bit

```
1. void Nhap(int &x)
2. {
3.     cin >> x;
4. }
```

```
5. void Xuat(int x)
6. {
7.     cout << x;
8. }
```

Câu 3 – Tổng bit

```
1. int TinhTongBit(int x)
2. {
3.     int tongBit = 0;
4.
5.     while (x > 0)
6.     {
7.         tongBit = tongBit + x % 2;
8.         x = x / 2;
9.     }
10.
11.     return tongBit;
12. }
```

Câu 4 – Có chứa số âm ?

- Viết hàm kiểm tra mảng có chứa số âm hay không

cuu duong than cong . com

cuu duong than cong . com

Câu 4 – Có chứa số âm ?

```
1. bool ChuaSoAm(float a[100], int n)
2. {
3.     bool chuaSoAm = false;
4.
5.     for (int i = 0; i < n; i++)
6.         if (a[i] < 0)
7.             chuaSoAm = true;
8.
9.     return chuaSoAm;
10. }
```

Câu 5 – Chu vi tam giác

- Viết chương trình nhập vào một tam giác gồm 3 đỉnh. Tính chu vi của tam giác đó và xuất kết quả.

cuu duong than cong . com

cuu duong than cong . com

Câu 5 – Chu vi tam giác

```
1. #include <iostream>
2. using namespace std;
3. struct DIEM
4. {
5.     float x;
6.     float y;
7. };
8. struct TAM_GIAC
9. {
10.    DIEM A;
11.    DIEM B;
12.    DIEM C;
13.};
```

Câu 5 – Chu vi tam giác

1. `void` Nhap(`DIEM`& diem);
2. `void` Nhap(`TAM_GIAC`& tg);
3. `float` TinhKhoangCach(`DIEM` A, `DIEM` B);
4. `float` TinhChuVi(`TAM_GIAC` tg);

cuu duong than cong . com

Câu 5 – Chu vi tam giác

```
1. void main()  
2. {  
3.     TAM_GIAC tg;  
4.     float chuVi;  
5.  
6.     cout << "Nhap tam giac" << endl;  
7.     Nhap(tg);  
8.  
9.     chuVi = TinhChuVi(tg);  
10.  
11.    cout << "Chu vi" << chuVi;  
12. }
```

Câu 5 – Chu vi tam giác

```
1. void Nhap(DIEM& D)
2. {
3.     cout << "Nhap hoành do = ";
4.     cin >> D.x;
5.     cout << "Nhap tung do = ";
6.     cin >> D.y;
7. }
```

cuu duong than cong . com

Câu 5 – Chu vi tam giác

```
1. void Nhap(TAM_GIAC& tg)
2. {
3.     cout << "Nhap dinh A = " << endl;
4.     Nhap(tg.A);
5.     cout << "Nhap dinh B = " << endl;
6.     Nhap(tg.B);
7.     cout << "Nhap dinh C = " << endl;
8.     Nhap(tg.C);
9. }
```

Câu 5 – Chu vi tam giác

```
1. float TinhKhoangCach(DIEM A, DIEM B)
2. {
3.     float khoangCach;
4.
5.     khoangCach = sqrt((A.x-B.x)*(A.x-B.x) +
6.                        (A.y-B.y)*(A.y-B.y));
7.     return khoangCach;
8. }
```

Câu 5 – Chu vi tam giác

```
1. float TinhChuVi(TAM_GIAC tg)
2. {
3.     float chuVi, a, b, c;
4.
5.     a = TinhKhoangCach(tg.B, tg.C);
6.     b = TinhKhoangCach(tg.A, tg.C);
7.     c = TinhKhoangCach(tg.A, tg.B);
8.     chuVi = a + b + c;
9.
10.    return chuVi;
11. }
```

Câu 6 – Sắp mảng tăng

- Viết chương trình đọc n số nguyên từ một tập tin MANG.INP, sau đó sắp xếp tăng dần rồi ghi kết quả vào 1 tập tin khác MANG_TANG.OUT. Ví dụ:

MANG.INP	MANG.OUT
4	4
2 5 1 4	1 2 4 5

Câu 6 – Sắp mảng tăng

1. `#include <stdio.h>`

2.

3. `void Nhap(char tenTapTin[100],
 int a[100], int &n);`

4. `void SapXep(int a[100], int n);`

5. `void Xuat(char tenTapTin[100],
 int a[100], int n);`

Câu 6 – Sắp mảng tăng

```
1. void main()  
2. {  
3.     int a[100], n;  
4.     Nhap("MANG.INP", a, n);  
5.     SapXep(a, n);  
6.     Xuat("MANG.OUT", a, n);  
7. }
```


Câu 6 – Sắp mảng tăng

```
1. void Nhap( char tenTapTin[100],  
              int a[100], int &n)  
2. {  
3.     FILE *f = fopen(tenTapTin, "r");  
4.     fscanf(f, "%d", &n);  
5.     for (int i = 0; i < n; i++)  
6.         fscanf(f, "%d", &a[i]);  
7.     fclose(f);  
8. }
```

Câu 6 – Sắp mảng tăng

```
1. void SapXep(int a[100], int n)
2. {
3.     for (int i = 0; i < n - 1; i++)
4.         for (int j = i + 1; j < n; j++)
5.             if (a[i] > a[j])
6.                 {
7.                     int tam = a[i];
8.                     a[i] = a[j];
9.                     a[j] = tam;
10.                }
11. }
```

Câu 6 – Sắp mảng tăng

```
1. void Xuat( char tenTapTin[100],  
              int a[100], int n)  
2. {  
3.     FILE *f = fopen(tenTapTin, "w");  
4.     fprintf(f, "%d\n", n);  
  
5.     for (int i = 0; i < n; i++)  
6.         fprintf(f, "%d\t", a[i]);  
7.     fclose(f);  
8. }
```