

## PHẦN 1: ÔN TẬP

Trong C++:

- Câu lệnh để khai báo biến có cấu trúc như sau: kiểu biến tên các biến;
  - Ví dụ 1: int a,b; Khai báo hai biến a,b có kiểu số nguyên.
  - Ví dụ 2: double a,b; Khai báo hai biến a,b có kiểu số thực.
  - Ví dụ 3: char a,b; Khai báo hai biến a,b có kiểu kí tự.
  - Ví dụ 4: string a,b; Khai báo hai biến a,b có kiểu chuỗi kí tự.
- Mỗi kiểu biến có các phép toán khác nhau:
  - Kiểu số nguyên có 5 phép toán: Cộng (+); Trừ(-); Nhân(\*); chia lấy nguyên (/); chia lấy dư (%)
  - Kiểu số thực có 4 phép toán: Cộng (+); Trừ(-); Nhân(\*); chia (/) ( **kiểu số thực không có phép chia lấy dư, chia lấy nguyên**)
  - Để lấy phần nguyên của số thực a, ta dùng câu lệnh int(a)**
  - Kiểu kí tự phép cộng, trừ hai kí tự là phép cộng, trừ của hai mã ASCII kí tự đó
  - Muốn **đổi kí tự ra mã ASCII ta dùng lệnh int(kí tự)**, ngược lại muốn đổi **mã ra kí tự ta dùng char(mã số)**. Ví dụ sau đây giúp ta đổi một kí tự thường ra kí tự hoa.  

```
char a;
cin>>a;
cout<<char(a-32);
```

### BẢNG MÃ ASCII

ASCII table upload by nguyenvanquan7826

0	20	40	60	80	100	120	140	160	180	200	220	240
1	21	41	61	81	101	121	141	161	181	201	221	241
2	22	42	62	82	102	122	142	162	182	202	222	242
3	23	43	63	83	103	123	143	163	183	203	223	243
4	24	44	64	84	104	124	144	164	184	204	224	244
5	25	45	65	85	105	125	145	165	185	205	225	245
6	26	46	66	86	106	126	146	166	186	206	226	246
7	27	47	67	87	107	127	147	167	187	207	227	247
8	28	48	68	88	108	128	148	168	188	208	228	248
9	29	49	69	89	109	129	149	169	189	209	229	249
10	30	50	70	90	110	130	150	170	190	210	230	250
11	31	51	71	91	111	131	151	171	191	211	231	251
12	32	52	72	92	112	132	152	172	192	212	232	252
13	33	53	73	93	113	133	153	173	193	213	233	253
14	34	54	74	94	114	134	154	174	194	214	234	254
15	35	55	75	95	115	135	155	175	195	215	235	255
16	36	56	76	96	116	136	156	176	196	216	236	
17	37	57	77	97	117	137	157	177	197	217	237	
18	38	58	78	98	118	138	158	178	198	218	238	
19	39	59	79	99	119	139	159	179	199	219	239	

- Khi gặp một bài toán ta phải thực hiện các bước sau:
  - Đọc kĩ đầu bài xác định INPUT; OUTPUT (Vào ; Ra)
  - Tìm kiểu dữ liệu phù hợp để khai báo dữ liệu vào ra
  - Tìm cách làm (thuật toán)
  - Viết chương trình
  - Chạy thử trên máy cá nhân nếu sai chính tả hoặc kết quả tự xem lại mình viết sai chỗ nào? Cách làm sai ở đâu và tự sửa
  - Nộp và chấm thử, nếu chưa đủ điểm thì kiểm tra lại cách làm

### Cấu trúc chung chương trình:

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
    freopen(".....", "r", stdin);
    freopen(".....", "w", stdout);
    kiểu biến tên biến;
    cin>>.....;
    các câu lệnh điều khiển cách làm;
    cout<<.....
```

```
    return 0;  
}
```

## CÁC CÂU LỆNH ĐIỀU KHIỂN CHƯƠNG TRÌNH

### 1. LỆNH IF

- **Dạng 1: Dạng khuyết**

if ( điều kiện) câu lệnh – Dùng trong trường hợp sau if chỉ có 1 lệnh duy nhất

hoặc if ( điều kiện) {khối lệnh } – Dùng trong trường hợp sau if chỉ có nhiều lệnh.

Trong đó: điều kiện là một biểu thức so sánh. Có 6 loại so sánh: >; <; >=; <=; == (BẰNG) ; != (KHÁC)

Tương ứng với các so sánh: Lớn hơn, nhỏ hơn, lớn hơn hoặc bằng, nhỏ hơn hoặc bằng, bằng, khác. Nếu trong điều kiện ta dùng nhiều điều kiện thì ta dùng phép toán và ( &&); hoặc (||). Ví dụ if((a>b) && (b==c) || (c<=d))

.. nghĩa là nếu a lớn hơn b và b bằng c hoặc c nhỏ hơn hoặc bằng d

Khi gặp câu lệnh này máy kiểm tra điều kiện. Nếu điều kiện đúng thực hiện câu lệnh sau if, sai thì không thực hiện câu lệnh sau if.

- **Dạng 2: Dạng đủ**

if ( điều kiện) câu lệnh1; else câu lệnh 2;

– Dùng trong trường hợp sau if chỉ có 1 lệnh duy nhất

hoặc if ( điều kiện) {khối lệnh1;}; else {khối lệnh2;};

– Dùng trong trường hợp sau if chỉ có nhiều lệnh.

Trong đó: điều kiện là một biểu thức so sánh. Có 6 loại so sánh: >; <; >=; <=; ==; !=

Tương ứng với các so sánh: Lớn hơn, nhỏ hơn, lớn hơn hoặc bằng, nhỏ hơn hoặc bằng, bằng, khác. Nếu trong điều kiện ta dùng nhiều điều kiện thì ta dùng phép toán và ( &&); hoặc (||). Ví dụ if((a>b) && (b==c) || (c<=d))

.. nghĩa là nếu a lớn hơn b và b bằng c hoặc c nhỏ hơn hoặc bằng d

Khi gặp câu lệnh này máy kiểm tra điều kiện. Nếu điều kiện đúng thực hiện câu lệnh 1 sau if, sai thì thực hiện câu lệnh 2 sau else

### 2. CÂU LỆNH LẶP VỚI SỐ LẦN BIẾT TRƯỚC

Lệnh for: Lệnh for là lệnh dùng để thực hiện các câu lệnh lặp đi lặp lại với số lần biết trước

Cú pháp của lệnh for như sau:

**for (variable\_initialization; condition; variable\_update)**

```
{
    statements;
}
```

Ví dụ :

**for (int count = 1; count <= 10; count++)**

```
{
    cout << count << endl;
}
```

Vòng lặp for được định nghĩa bởi từ khóa **for** và được chia làm 3 phần chính, mỗi phần được ngăn cách bởi dấu chấm phẩy:

- **Variable initialization (phần khởi tạo biến)**

Biến vòng lặp có thể được khai báo và khởi tạo giá trị ngay bên trong phần khởi tạo của vòng lặp **for**.

Như ở ví dụ trên, biến count được khai báo và khởi tạo với giá trị 1.

Phần khởi tạo biến được thực thi đầu tiên và chỉ thực thi 1 lần duy nhất trong vòng lặp **for**.

- **Condition (biểu thức điều kiện)**

Khối lệnh của vòng lặp for sẽ được thực hiện nếu biểu thức điều kiện cho giá trị đúng. Vòng lặp **for** kiểm tra biểu thức điều kiện trước khi thực hiện khối lệnh.

- **Variable update (cập nhật biến vòng lặp)**

Phần này sẽ được thực thi cuối mỗi lần lặp, sau khi khối lệnh của vòng lặp **for** được thực thi. Phần này thường chịu trách nhiệm thay đổi giá trị biến vòng lặp được sử dụng trong biểu thức điều kiện (nhằm tránh tình trạng lặp vô hạn). Sau khi thực thi xong phần cập nhật biến vòng lặp, chương trình quay trở lại đánh giá biểu thức điều kiện của vòng lặp **for** và cứ như thế.

- ```
{
    statements;
}
```

Phần nằm trong cặp ngoặc nhọn trên là khối lệnh ( có thể 1 hoặc nhiều lệnh) cần thực hiện. Chẳng hạn ở ví dụ trên chương trình sẽ viết các số từ 1 đến 10, mỗi số trên một dòng. Áp dụng lệnh for làm các bài tập sau:

### 3. LẶP VỚI SỐ LẦN CHƯA BIẾT TRƯỚC

while (điều kiện) câu lệnh; hoặc while (điều kiện) {khối câu lệnh};

- **Phần điều kiện gống lệnh if**

- **Cách thực hiện:** máy kiểm tra điều kiện nếu đúng thì thực hiện câu lệnh sau điều kiện (khối lệnh sau điều kiện) sau đó máy lại quay lại kiểm tra nếu điều kiện sai thì dừng



## PHẦN 2: BÀI TẬP

### Bài 1: Tổng cuối

Nhập vào 1 số tự nhiên N. Tính tổng 2 chữ số cuối của số N. Nếu N không đủ 2 chữ số thì in ra 0  
Ví dụ:

| TONGCUOI.INP | TONGCUOI.OUT |
|--------------|--------------|
| 12           | 3            |
| 123          | 5            |
| 2            | 0            |

**Bài 2: Số nhà** Ucoder trên đường đi học về đi qua một con phố có N ngôi nhà được đánh số từ 1 tới N. Những ngôi nhà bên trái của dãy phố được đánh số lẻ, ở bên phải được đánh số chẵn. Ucoder vừa đi vừa nhìn sang bên trái và tính nhẩm tổng các số nhà trên con phố.

Bạn hãy giúp Ucoder tính tổng này nhé.

Mô tả đầu vào

Dòng duy nhất chứa số tự nhiên N.

Ràng buộc  $0 < N < 1000000000$

Ví dụ:

| SONHA.INP | SONHA.OUT |
|-----------|-----------|
| 5         | 9         |
| 18        | 81        |

### Bài 3: Tensdigit

Nhập vào số tự nhiên N. In ra chữ số hàng chục của nó. Nếu không có ghi -1

Ví dụ:

| TENSDIGIT.INP | TENSDIGIT.OUT |
|---------------|---------------|
| 5             | -1            |
| 18            | 1             |

### Bài 4: Tiền điện

Lượng điện năng tiêu thụ hàng tháng của gia đình Ucoder được tính bằng số điện (kW).

Biết giá tiền cho 100 số điện đầu tiên là 3000 VND mỗi số, từ số điện thứ 101 trở lên, giá là 4500 VND.

Tháng này, nhà Ucoder sử dụng hết N số điện, hỏi tổng tiền bố Ucoder cần trả cho công ty điện lực là bao nhiêu?

| TIENDIEN.INP | TIENDIEN.OUT |
|--------------|--------------|
| 10           | 30000        |
| 120          | 390000       |

### Bài 5: Máy tính

Trong một buổi học toán, thầy giáo tổ chức trò chơi tính nhẩm.

Thầy giáo có một tập các thẻ số và một tập các thẻ phép tính +, -, \*, / tương ứng với các phép toán cộng, trừ, nhân, chia.

Hai bạn Ucoder và Ucodi được thầy gọi lên bảng, Ucoder cầm tập các thẻ số, Ucodi cầm các thẻ phép tính. Thầy giáo yêu cầu bạn Ucoder giơ lên ba số bất kì lần lượt là a, b và n, bạn Ucodi phải giơ tấm thẻ phép tính phù hợp để điền vào ô trống sau:  $a \square b = n$  sao cho thu được được phép tính đúng.

Cho ba số a, b, n, Ucodi hãy tìm phép tính phù hợp, nếu không có phép tính phù hợp, Ucodi phải giơ tấm thẻ NO.

Dòng duy nhất là đáp án của bài toán.

Ví dụ:

| MAYTINH.INP | MAYTINH.OUT |
|-------------|-------------|
| 8 8 1       | /           |

**Bài 6:** Nhập vào một kí tự, nếu là chữ số ghi chữ số đó, nếu không phải là kí tự in mã ASCII của kí tự đó

Ví dụ 1:

| NHAPKT.INP | NHAPKT.OUT |
|------------|------------|
| 2          | 2          |

Ví dụ 2:

|            |            |
|------------|------------|
| NHAPKT.INP | NHAPKT.OUT |
| A          | 65         |

**Bài 7:** Nhập vào một số, nếu số đó là số thực thì ghi phần nguyên của nó, nếu là số nguyên kiểm tra xem số đó là số chẵn hay lẻ

Ví dụ 1:

|            |              |
|------------|--------------|
| NHAPSO.INP | NHAPKT.OUT   |
| 2          | 2 là so chan |

Ví dụ 2:

|            |            |
|------------|------------|
| NHAPSO.INP | NHAPKT.OUT |
| 3          | 3 là so le |

Ví dụ 3:

|            |            |
|------------|------------|
| NHAPSO.INP | NHAPSO.OUT |
| 2.55       | 2          |

**BÀI 8:** Nhập vào số N, nếu N là số chẵn thì ghi các số chẵn từ 0 đến N, nếu N lẻ ghi các số lẻ từ 1 đến N

Ví dụ 1:

|            |            |
|------------|------------|
| INSOCL.INP | INSOCL.OUT |
| 6          | 0 2 4 6    |

Ví dụ 2:

|            |            |
|------------|------------|
| INSOCL.INP | INSOCL.OUT |
| 7          | 1 3 5 7    |

**Bài 9:** Ước chung lớn nhất của hai số a và b là số lớn nhất mà a và b cùng chia hết cho số đó. Ví dụ: số 30 và số 45 cùng chia hết cho 1; 5; 15 thì 15 là ước chung lớn nhất của 30 và 45. Để tìm ước chung lớn nhất của a và b có nhiều cách, tuy nhiên cách nhanh nhất là sử dụng thuật toán do nhà toán học Ôclit tìm ra cách đây hơn 4000 năm. Thuật toán tìm UCLN của hai số a và b như sau:

Bước 1: Nếu b=0 thì ước chung lớn nhất là a

Bước 2: a=b; b=r (r là số dư của a chia cho b) quay lại bước 1

Ví dụ: Tìm UCLN(72, 56)=UCLN(56,16)=UCLN(16,8)=UCLN(8,0)=8

Nhập vào hai số a và b, lập trình tìm UCLN theo thuật toán trên

Ví dụ:

|           |           |
|-----------|-----------|
| OCLIT.Inp | OCLIT.OUT |
| 45 30     | 15        |

**Bài 10:** Cho hai số a và b, nếu a chia hết cho b ghi thương a chia cho b ngược lại ghi phân số a/b dưới dạng tối giản

Ví dụ 1:

|            |            |
|------------|------------|
| PHANSO.INP | PHANSO.OUT |
| 6 3        | 2          |

Ví dụ 2:

|            |            |
|------------|------------|
| PHANSO.INP | PHANSO.OUT |
| 6 4        | 3/2        |

### **Bài 11 Xếp khách**

Một khách sạn có N phòng đôi (phòng cho 2 người) được đánh số từ 1 đến N. Khi có một đoàn khách đến thuê phòng, tiếp tân của khách sạn sẽ xếp khách vào phòng theo quy tắc sau: Mỗi cặp khách sẽ được xếp vào phòng chỉ số nhỏ nhất trong các số phòng trống. Nếu số lượng khách của đoàn là số lẻ thì người cuối cùng của đoàn khách sẽ được xếp vào phòng có chỉ số nhỏ nhất trong số các phòng trống. Nếu không còn phòng trống, thì số khách chưa có phòng sẽ được xếp tuần tự từng người một vào phòng có chỉ số nhỏ nhất trong số các phòng mới có một khách ở (của đoàn khách đến trước). Đầu tiên tất cả các phòng của khách sạn là trống.

Yêu cầu: Cho trước trình tự đến của các đoàn khách và số lượng khách của mỗi đoàn. Hãy xác định số lượng khách trong mỗi phòng của khách sạn.

Dữ liệu: Vào từ file văn bản ROOMS.INP:

- Dòng đầu tiên chứa 2 số nguyên dương  $N$  ( $1 \leq N \leq 100$ ) và  $G$  được ghi cách nhau bởi dấu cách.  $N$  là số phòng của khách sạn,  $G$  là số lượng đoàn khách.

- Dòng thứ  $i$  trong số  $G$  dòng tiếp theo chứa số lượng khách của đoàn khách thứ  $i$  (các đoàn khách được đánh số theo thứ tự đến khách sạn bắt đầu từ 1).

Giả thiết rằng không có 2 đoàn khách nào đến khách sạn vào cùng một thời điểm và tổng số khách của tất cả các đoàn không vượt quá sức chứa của khách sạn ( $\leq 2N$ ).

Kết quả: Ghi ra file ROOMS.OUT  $N$  dòng: dòng thứ  $i$  chứa số lượng khách của phòng  $i$  ( $1 \leq i \leq N$ ).

Ví dụ:

| ROOMS.INP | ROOMS.OUT |
|-----------|-----------|
| 7 3       | 2         |
| 3         | 1         |
| 1         | 1         |
| 4         | 2         |
|           | 2         |
|           | 0         |
|           | 0         |