

Lab 02

A. Mục tiêu

- SV hiểu và sử dụng được các kiểu dữ liệu cơ bản
- SV tạo và sử dụng được biến của các kiểu cơ bản để tính toán
- Giới thiệu lệnh if
- Tạo số ngẫu nhiên

B. Bài mẫu²

Đề bài: Viết chương trình cho biết số nguyên đọc vào từ bàn phím là chẵn hay lẻ.

Cách giải 1: Sử dụng phép chia lấy dư

```
1  #include<stdio.h>
2
3  int main(){
4      int n;
5      printf("Nhap mot so nguyen: ");
6      scanf("%d",&n);
7
8      if(n % 2 == 0)
9          printf("%d la so chan\n", n);
10     else
11         printf("%d la so le\n", n);
12
13     return 0;
14 }
```

Câu lệnh if dạng đơn giản có dạng:

```
if ([biểu_thức_logic])
    [thực hiện công việc 1];

else
    [thực hiện công việc 2];
```

Dòng 8 đến 11 có nghĩa là: nếu biểu thức $(n \% 2 == 0)$ đúng thì thực hiện dòng 9, ngược lại thực hiện dòng 11.

Cách giải 2 (tham khảo): Sử dụng phép toán trên bit,

```
1  #include<stdio.h>
2
3  int main(){
4      int n;
5      printf("Nhap mot so nguyen: ");
6      scanf("%d",&n);
7
8      if(n & 1 == 1)
9          printf("%d la so le\n", n);
10     else
11         printf("%d la so chan\n", n);
12
13     return 0;
14 }
```

Biểu thức logic của lệnh if ở dòng 8 có nghĩa là nếu bit cuối cùng của số nguyên n là 1 thì n là số lẻ.

Cách giải 3 (tham khảo): Sử dụng toán tử điều kiện (một số tài liệu gọi là phép toán tam nguyên)

```
#include<stdio.h>

int main(){
    int n;
    printf("Nhập một số nguyên: "); scanf("%d",&n);

    n%2 == 0 ? printf("%d là số chẵn\n", n) : printf("%d là số lẻ\n", n);

    return 0;
}
```

Toán tử điều kiện có dạng [biểu_thức_logic] ? [công_việc_1] : [công_việc_2];

Nếu [biểu_thức_logic] đúng thì [công_việc_1] sẽ được thực hiện, ngược lại [công_việc_2] sẽ được thực hiện.

Cách giải 4 (tham khảo): Không sử dụng phép toán trên bit hay phép chia lấy dư

```
#include<stdio.h>

int main(){
    int n;
    printf("Nhập một số nguyên: "); scanf("%d",&n);

    if((n / 2) * 2 == n)
        printf("%d là số chẵn\n", n);
    else
        printf("%d là số lẻ\n", n);

    return 0;
}
```

C. Bài tập

Sinh viên đặt tên mỗi bài theo cấu trúc: Lab2_B1.C, Lab2_B2.C....

Sau đó cho vào 1 thư mục, nén với tên: MSV.zip/zar

- Viết chương trình chuyển đổi một số nguyên từ hệ thập phân sang số nhị phân và in ra màn hình.
- Viết chương trình chuyển đổi một số nhị phân sang số nguyên thập phân và in ra màn hình.
- Viết chương trình tạo 5 số ngẫu nhiên chỉ sử dụng hàm **rand()**, chạy vài lần để kiểm chứng, sau đó bổ sung thêm hàm **srand(seed)**, chạy vài lần và kiểm chứng lại xem sự khác biệt là gì?

4. Sinh ra một chuỗi số nguyên (100 số) ngẫu nhiên trong khoảng $[a, b]$ bất kỳ, nếu muốn in ra chuỗi số 0, 1 thì thay đổi thế nào? Hãy in ra số byte được sử dụng để chứa các số ngẫu nhiên được tạo ra.
5. Tạo 2 số thực ngẫu nhiên x_1, x_2 thỏa mãn $0 < x_1, x_2 < 1$. Làm tròn số x theo yêu cầu sau
- Nếu $0 < x < 0.5$ thì làm tròn $x=0$
 - Nếu $0.5 \leq x < 1$ thì làm tròn $x=1$
6. Tạo 1 số thực ngẫu nhiên x thỏa mãn $a < x < b$. Làm tròn số x theo yêu cầu sau
- Làm tròn sau dấu phẩy 1 chữ số thập phân và in ra lấy 2 chữ số thập phân
Ví dụ: 6.796686 làm tròn thành 6.8 và in ra 6.80
 - Làm tròn sau dấu phẩy 2 chữ số thập phân và in ra lấy 3 chữ số thập phân
Ví dụ: 5.914579 làm tròn thành 5.91 và in ra 5.910
 - In kết quả của số vừa tạo chia cho 1000000 dưới dạng dấu phẩy động có số mũ
Ví dụ: 5.93e-003
7. Thực hiện tạo 2 số ngẫu nhiên x, n thỏa mãn $-3 < x < 3$ và $0 < n < 1$. Làm tròn số $y = x+n$ theo yêu cầu như dưới, in ra với độ rộng 7 ký tự và lấy 2 chữ số thập phân.
- Nếu $0 < \text{phần thập phân} < 0.5$ thì làm tròn xuống
 - Nếu $0.5 \leq \text{phần thập phân} < 1$ thì làm tròn lên