



ĐẠI HỌC BÁCH KHOA HÀ NỘI  
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

# C PROGRAMMING INTRODUCTION

## TUẦN 8: VÒNG LẶP

# Nội dung

- Vòng lặp
  - Ôn tập
    - While, do... while
    - Chú ý
    - Continue và break
  - Thực hành

# Cấu trúc lặp while, do

- Câu lệnh while
  - Biểu thức được kiểm tra. Nếu *true*, câu lệnh được thực hiện và biểu thức được kiểm tra lại. Vòng lặp tiếp tục tới khi biểu thức *false*.

```
while (expression) {  
    Statement1;  
    Statement2;  
    ...  
}
```

# Cấu trúc lặp while, do

- VD While

```
#include <stdio.h>
#define PERIOD '.'
main() {
    char C;
    while ((C = getchar()) != PERIOD)
        putchar(C);
    printf("Good Bye.\n");
}
```

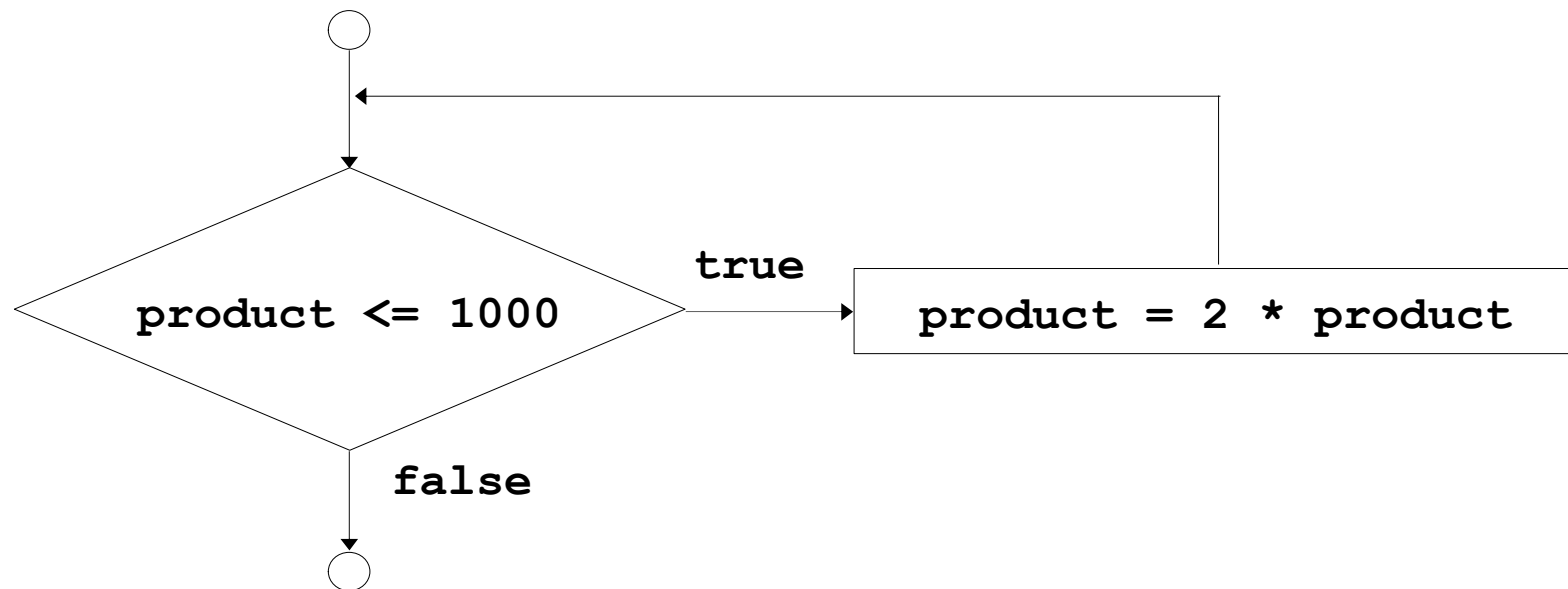


**Result?**

# Cấu trúc lặp while, do

- VD:

```
int product = 2;  
while ( product <= 1000 )  
    product = 2 * product;
```



# Cấu trúc lặp while, do

- Câu lệnh do-While
  - Do-while kiểm tra điều kiện sau khi thực hiện câu lệnh; Câu lệnh luôn được thực hiện ít nhất một lần

```
do {  
    statement1;  
    statement2;  
    ...  
} while (expression);
```

# Cấu trúc lặp while, do

- VD Do-While

```
int i = 1, sum = 0;  
do {  
    sum += i;  
    i++;  
} while (i <= 50);  
printf("The sum of 1 to 50 is %d\n", sum);
```



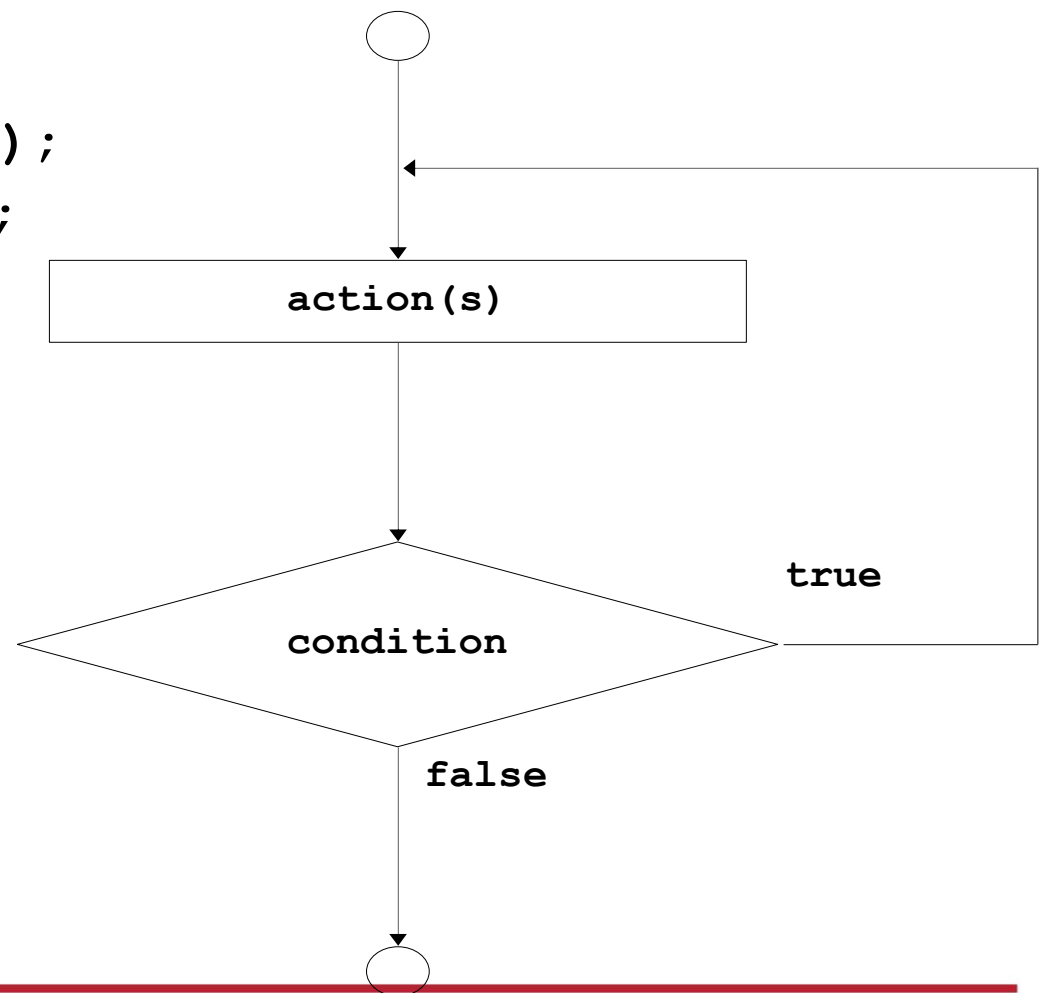
Result?

# Cấu trúc lặp while, do

- V (letting **counter** = 1)

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

In các số nguyên từ 1 đến 10





# Continue và Break

- Câu lệnh Break và Continue

- Câu lệnh **break** thoát ra khỏi vòng lặp while và do.

**break ;**

- Câu lệnh **continue** thực hiện vòng lặp tiếp theo mà bỏ qua các câu lệnh bên dưới bên trong vòng lặp hiện tại

**continue ;**

# Continue và Break

- VD Break và Continue

```
int c;  
while ((c = getchar()) != -1) {  
    if (C == '.')  
        break;  
    else if (c >= '0' && c <= '9')  
        continue;  
    else putchar(c);  
}  
printf("*** Good Bye ***\n");
```

# Bài tập 8.1

- Viết chương trình in ra nội dung người dùng nhập vào từ bàn phím thay thế chuỗi liên tiếp dấu cách ‘ ‘ bằng một dấu cách duy nhất
- Có thể sử dụng `getchar()` và `putchar()`

# Lời giải

```
#include <stdio.h>
int main(void)
{
    int c;
    int inspace;
    inspace = 0;
    while((c = getchar()) != EOF)
    {
        if(c == ' ')
        {
            if(inspace == 0)
            {
                inspace = 1;
                putchar(c);
            }
        }
    }
}
```

# Lời giải

```
}
```

```
/* We haven't met 'else' yet, so we have to be a little clumsy */
```

```
if(c != ' ')
```

```
{
```

```
    inspace = 0;
```

```
    putchar(c);
```

```
}
```

```
}
```

```
return 0;
```

```
}
```

# Bài tập 8.2

- Viết chương trình thay thế các kí tự ‘\’, ‘\t’, ‘\b’ bởi ‘\\’, ‘\\t’, ‘\\b’ trong chuỗi nhập vào và in ra màn hình.
- Có thể sử dụng hàm `getchar()`
- Sử dụng *if* hoặc *switch*

# Lời giải

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int c, d;
```

```
    while ( (c=getchar()) != EOF) {
```

```
        d = 0;
```

```
        if (c == '\\') {
```

```
            putchar('\\');
```

```
            putchar('\\');
```

```
            d = 1;
```

```
        }
```

# Lời giải

```
if (c == '\t') {  
    putchar('\\');  
    putchar('t');  
    d = 1;  
}  
if (c == '\b') {  
    putchar('\\');  
    putchar('b');  
    d = 1;  
}  
if (d == 0)  
    putchar(c);  
}  
return 0;  
}
```



# Bài tập 8.3

- Tính lương cho nhân viên
- Viết và biên dịch chương trình sau và giải thích kết quả.

# exercise8\_3.c

```
#include <stdio.h>
```

```
int
```

```
main(void)
```

```
{
```

```
    double total_pay; /* company payroll */
```

```
    int count_emp; /* current employee */
```

```
    int number_emp; /* number of employees */
```

```
    double hours; /* hours worked */
```

```
    double rate; /* hourly rate */
```

```
    double pay; /* pay for this period */
```

```
    /* Get number of employees. */
```

```
    printf("Enter number of employees> ");
```

```
    scanf("%d", &number_emp);
```

```
/* Compute each employee's pay and add it to the payroll. */
total_pay = 0.0;
count_emp = 0;
while (count_emp < number_emp) {
    printf("Hours> ");
    scanf("%lf", &hours);
    printf("Rate > $");
    scanf("%lf", &rate);
    pay = hours * rate;
    printf("Pay is $%6.2f\n\n", pay);
    total_pay = total_pay + pay;
    count_emp = count_emp + 1;
}
printf("All employees processed\n");
printf("Total payroll is $%8.2f\n", total_pay);
return (0);
}
```

# Bài tập 8.4

- Viết chương trình sử dụng vòng lặp *while* để đếm số sinh viên qua và số sinh viên trượt.
- Yêu cầu người dùng nhập vào đánh giá: 1 : qua và 2 : trượt.

# Lời giải

```
#include <stdio.h>

/* function main begins program execution */
int main( void )
{
    /* initialize variables in definitions */
    int passes = 0; /* number of passes */
    int failures = 0; /* number of failures */
    int student = 1; /* student counter */
    int result;      /* one exam result */

    /* process 10 students using counter-controlled loop */
    while ( student <= 10 ) {

        /* prompt user for input and obtain value from user */
        printf( "Enter result ( 1=pass,2=fail ): " );
        scanf( "%d", &result );
```

# Lời giải

```
/* if result 1, increment passes */
    if ( result == 1 ) {
        passes = passes + 1;
    } /* end if */
    else { /* otherwise, increment failures */
        failures = failures + 1;
    } /* end else */

    student = student + 1; /* increment student counter */
} /* end while */

/* termination phase; display number of passes and failures */
printf( "Passed %d\n", passes );
printf( "Failed %d\n", failures );
return 0; /* indicate program ended successfully */

} /* end function main */
```

# Exercise 8.5

- Sử dụng do...while để in ra các số nguyên nhỏ hơn một số cho trước.

# Lời giải

```
#include <stdio.h>

/* function main begins program execution */
int main( void )
{
    int counter = 1;          /* initialize counter */

    do {
        printf( "%d ", counter ); /* display counter */
    } while ( ++counter <= 10 ); /* end do...while */

    return 0; /* indicate program ended successfully */

} /* end function main */
```



# Bài tập 8.6

- Tính điểm trung bình
- Gợi ý:
  - Lưu vết tổng điểm tích lũy và số điểm
  - Cần tiếp tục đọc đến khi gặp dấu hiệu kết thúc (sử dụng điểm  $< 0$ )
  - In ra kết quả

# Lời giải sử dụng `while`

```
# include <stdio .h>
int main ()
{
    float grade , sum = 0.0;
    int gradeCount = 0;
    printf (" Enter grade : ");
    scanf ("%g", & grade );
    while ( grade >= 0.0) {
        sum += grade ;
        ++ gradeCount ;
        printf (" Enter grade : ");
        scanf ("%g", & grade );
    }
    printf (" Average : %g\n",
sum/ gradeCount );
    return 0;
}
```

# Lời giải sử dụng do...while

```
# include <stdio .h>
int main () {
    float grade , sum;
    int gradeCount ;
    int another ;
    do {
        sum = gradeCount = 0;
        printf (" Enter grade : ");
        scanf ("%g", & grade );
        while ( grade >= 0.0) {
            sum += grade ;
            ++ gradeCount ;
            printf (" Enter grade : ");
            scanf ("%g", & grade );
        }
        printf (" Average : %g\n\n",
            sum/ gradeCount );
        printf (" Another class : ");
        scanf ("%d", & another );
    } while ( another != 0);
    return 0;
}
```

# Bài tập 8.7

- Viết chương trình tính  $n!$
- Sử dụng:
  - Biến đếm  $i$  chạy từ 1 tới  $n$ .
  - Biến kết quả cập nhật theo từng giá trị  $i$

# Lời giải

```
/* n! using while . */  
# include <stdio .h>  
int main () {  
    int i, n, f;  
    printf (" Enter n: ");  
    scanf ("%d", &n);  
    f = 1; /* 0! */  
    i = 1;  
    while (i <= n) {  
        f *= i; /* Now , f = i! */  
        ++i;  
    }  
    printf ("%d! = %d\n", n, f);  
    return 0;  
}
```



25 YEARS ANNIVERSARY  
**SOICT**

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG  
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

**Thank you  
for your  
attentions!**



[soict.hust.edu.vn/](http://soict.hust.edu.vn/)



[fb.com/groups/soict](https://fb.com/groups/soict)

