



# **Recitation Class 06 for VG101**

---

Date: 2012 / 10 / 29

Wang Qian

## feedback & announcement

- Please pay attention to the announcement about the resubmission policy.
- Take your flash disk during the lab time.
- LAB 04
  - Like a reading lab =\_ =
- HWK03
  - Most people did a good job, but please pay attention to the axis in Problem 3.



# Basic Concepts in C

- Some books I like (personally):
  - *The C Programming Language* (B.W. Kernighan & D.M. Ritchie)
  - *C++ Primer* (S.B. Lippman, J. Lajoie & B.E. Moo)
- Follow this problem for the whole class:
- Calculate the sum of the first n elements in a harmonic series:

$$S_n = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \dots + \frac{1}{n}$$

# Comments

- Two ways for you to add comments in C and C++
  - `/* ..... */` for code block
  - `//.....` for a single line
- E.g.

```

- /******
   * This code is to calculate the *
   * sum of harmonic series. >_<  *
   *****/

// Date: 2012-10-29
```

# Main Function

- The main function is a necessary part for any C or C++ function.
- It has a return value 0. (Recall the concept of return value in a function)
- Semi-colon is needed at the end of every expression.
- E.g.

```
int main()  
{  
    return 0;  
}
```



# Variables

- Must be declared before the expressions in C:
- Initialize with assigning operator =.
- Lifespan!
- Different types and sizes:
  - int            32 bits
  - short        16 bits
  - char         8 bits
  - float        32 bits
- E.g.

```
= int main() {  
    int n;  
    float ans = 0;
```

# Standard Input and Output

- Input:
  - Recall what a placeholder means in Matlab!
  - Do not forget to add & before the variable!
- Output:
  - Recall the usage of fprintf() in Matlab!
  - Recall what a conversion specification means in Matlab!
- E.g.
  - It seems that something goes wrong here T\_T

```
scanf ("%d", &n) ;  
printf ("The sum is: %f\n", ans) ;  
system ("pause") ;
```

# Library

- Include a library before use the functions in it.
  - printf() and scanf() can be used only after you include the library for standard input and output, <stdio.h>.
  - system("pause") can be used only after you include the standard library, <stdlib.h>.
- Libraries are included at the very beginning.
- E.g.

```
// Date: 2012-10-29

#include <stdio.h>
#include <stdlib.h>

int main() {
```



# Operators

- Most of them have the same meaning as in Matlab:
  - For example, `+` `-` `*` `/` `=` still have the same meaning.
  - `^` does not mean power in C or C++
  - `%` means modulo in C or C++
  - Bitwise operators may seem new to you, but they will be useful. They include: `~` , `&` , `|` , `^` , `<<` , `>>`.
  - Please compare them with logical operators: `!` , `&&` , `||`.
  - Increment and decrement operators. What the difference between `"n = i++;"` and `"n = ++i;"`?
  - Assignment operators may provide you with great convenience. They include: `+=` , `-=` , `*=` , `&=` , `<<=` and so on.
  - Ternary operator is also a very interesting one. For example, check `"a = 5 < 3 ? 1 : 0;"` and `"b = 4 > 2 ? 1 : 0;"`.

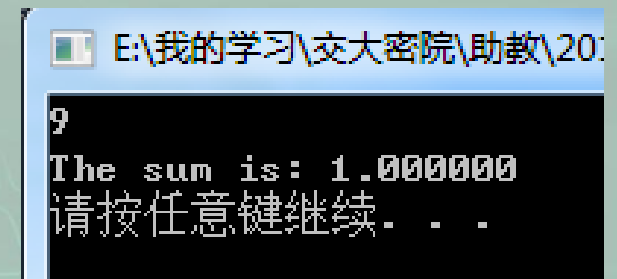
# For loop

- ```
for (i = 0; i < n; i++)  
{  
    expression;  
}
```
- Still the same meaning as in Matlab.
- Remember that variable *i* should also be declared.
- ```
for i = 0:1:(n-1)  
    expression  
end
```

# Debug

- What is wrong here -\_- ?

```
int main() {  
    int n, i;  
    float ans = 0;  
    scanf("%d", &n);  
    for (i = 1; i <= n; i++)  
        ans += 1/i;  
    printf("The sum is: %f\n", ans);  
    system("pause");  
    return 0;  
}
```



E:\我的学习\交大密院\助教\201...

9

The sum is: 1.000000

请按任意键继续...



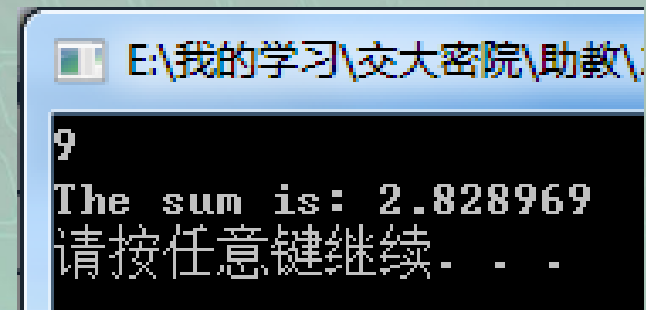
# Debug

- Build and compile Ctrl + Shift + B
- Start bebugging and continue F5
- Stop bebugging Shift + F5
- Run without bebugging Ctrl + F5
- Breakpoint F9
- Step into F11
- Step over F10
- Step out Shift + F11

# C type casting

- Let's discuss this expression:
  - `ans += 1/i;`
- What kind of type for `ans` and `i`?
- (type cast) variable
- Notice the priority! (Find on the lecture note)

```
int main() {  
    int n, i;  
    float ans = 0;  
    scanf("%d", &n);  
    for (i = 1; i <= n; i++)  
        ans += (float) 1/i;  
    printf("The sum is: %f\n", ans);  
    system("pause");  
    return 0;  
}
```



```
E:\我的学习\交大密院\助教\  
9  
The sum is: 2.828969  
请按任意键继续. . .
```

## Thinking more

- What about when  $n$  is very large?
  - e.g.  $n = 2000000000$
- We change float into double, “%f” to “%lf”...
- But it seems too slow!
- Use the following equation:

$$S_n = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \dots + \frac{1}{n} \approx \ln n + C$$



# Macro definition

- #define TOKEN value
- TOKEN represented in capitals.
- Material on SAKAI about coding style.

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

#define EULER 0.5772156649

int main() {
```

- Try this:
- #define A 3+5                      V.S.                      #define A (3+5)

# New Version

- Everything is OK now?

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

#define EULER 0.5772156649

int main()
{
    int n,i;
    double ans = 0;
    scanf("%d",&n);
    ans = log((double) n) + EULER;
    printf("The sum is: %lf\n",ans);
    system("pause");
    return 0;
}
```

E:\我的学习\交大密院\助教\201

2000000000

The sum is: 21.993629

请按任意键继续. . .

# New Version

- Are you kidding me?

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

#define EULER 0.5772156649

int main()
{
    int n,i;
    double ans = 0;
    scanf("%d",&n);
    ans = log((double) n) + EULER;
    printf("The sum is: %lf\n",ans);
    system("pause");
    return 0;
}
```

E:\我的学习\交大密院\助教\20

1

The sum is: 0.577216

请按任意键继续. . .



# If statement

- ```
if (condition 1) {  
    expression 1  
}  
else if (condition 2) {  
    expression 2  
}  
else (condition 3) {  
    expression  
}
```
- Use “else if” instead of “elseif”.
- Don’t forget the brackets.

# Function

- `return_type name (para_type para_name, ...)`  
  {  
    variable declarations;  
    expressions;  
    return value or variable;  
  }
- Notice that sometimes the `return_type` is `void`.
- At that time you just need “`return;`” at the end of the func.
- Or any time you want to quit this func.
- Prototype is needed.

# Function

- Prototype:

```
double calcHarmoSum(int n);
```

- Body:

```
double calcHarmoSum(int n)
{
    int i;
    double ans = 0;

    if (n > LMT)
        ans = log((double) n) + EULER;
    else
        for (i = 1; i <= n; i++)
            ans += (double) 1/i;

    return ans;
}
```



# Whole Picture

```

/*****
 * This code is to calculate the *
 * sum of harmonic series. >_< *
 *****/

// Date: 2012-10-29

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#define EULER 0.5772156649
#define LMT 1e6

double calcHarmoSum(int n);

```

```

int main()
{
    int n,i;
    double ans = 0;
    scanf("%d",&n);

    if (n > 0)
        ans = calcHarmoSum(n);
    else
    {
        printf("Input Error!\n");
        system("pause");
        return 0;
    }

    printf("The sum is: %lf\n",ans);
    system("pause");
    return 0;
}

```

```

double calcHarmoSum(int n)
{
    int i;
    double ans = 0;

    if (n > LMT)
        ans = log((double) n) + EULER;
    else
        for (i = 1; i <= n; i++)
            ans += (double) 1/i;

    return ans;
}

```

# Quiz

- Can you recognize what the following function means?

```
int lowbit(int x)
{
    return x & (x - 1) ^ x;
}
```

# Quiz

- Challenging Problem (beyond the course):
- This will never be covered in any assignment or exam, I think. But, I also firmly believe that the conclusion of this problem is useful.
- What will be shown after this expression?
  - `printf("%d %d", ++n, lowbit(n));`
- What is the value of `a[i-1]` and `a[i]` after this expression?
  - `a[i] = i++;`