SUSTech Southern University of Science and Technology

CS111，C Programming

Lab / Pointer

黄嘉炜

huangjw3@mail.sustech.edu.cn

深港微电子学院
SCHOOL OF MICROELECTRONICS

Outline

SUSTech Southern University of Science and Technology

- Review
- Pointer: basic intro & interesting case
- Pointer & Array
- Assignment

Review：Palindrom, why OJ canceled ?

SUSTech Southern University of Science and Technology

#6-3	✕ Cancelled	0ms	0 Bytes
#6-4	✕ Cancelled	0ms	0 Bytes
#6-5	✕ Cancelled	0ms	0 Bytes
#6-6	✕ Cancelled	0ms	0 Bytes
#6-7	✕ Cancelled	0ms	0 Bytes

Input

A single line containing the sentence to be checked. The sentence consists of less than 10^6 characters.

```
5       char str[100000];
6       fgets(str, 100000, stdin);
```

Review：Matrix Mul, OJ Time Exceeded ?

SUSTech Southern University of Science and Technology

Idea：calculated all element of C, then print out some part of it …

#16	✕ Time Exceeded	0	≥1101ms	≥11.4 MiB
#17	✕ Time Exceeded	0	≥1101ms	≥12.3 MiB
#18	✕ Time Exceeded	0	≥1102ms	≥13.1 MiB
#19	✕ Time Exceeded	0	≥1101ms	≥10.8 MiB
#20	✕ Time Exceeded	0	≥1102ms	≥13.1 MiB

Outline

SUSTech Southern University of Science and Technology

- Review
- **Pointer: basic intro & interesting case**
- Pointer & Array
- Assignment

Pointer: basic intro

SUSTech Southern University of Science and Technology

```
5       int val = 100;
6       printf("val %d, sizeof %u, at address: %p \n", val, sizeof(val), &val);
7
8       int* pval = &val;
9       printf("pval %x, sizeof %u, at address: %p \n", pval, sizeof(pval), &pval);
10
11      *pval = 200;
12      *pval ++;
13      printf("final val: %d \n", val);
14      printf("final pval: %x \n", pval);
```

{data_type} * {variable_name} ;

⇒ {data_type}: int / float / char / …
⇒ Why need data_type for pointer ?

Highlight

- 指针是一种特殊的变量，用来存储内存地址
- 指针运算：& (取变量的地址)，* (取地址的变量)

Pointer: basic intro

SUSTech Southern University of Science and Technology

```
5     int val = 100;
6     printf("val %d, sizeof %u, at address: %p \n", val, sizeof(val), &val);
7
8     int* pval = &val;
9     printf("pval %x, sizeof %u, at address: %p \n", pval, sizeof(pval), &pval);
10
11    *pval = 200;
12    *pval ++;
13    printf("final val: %d \n", val);
14    printf("final pval: %x \n", pval);
```

```
val 100, sizeof 4, at address: 000000000061FE1C
pval 61fe1c, sizeof 8, at address: 000000000061FE10
final val:
final pval:
```

??

Pointer: basic intro

SUSTech Southern University of Science and Technology

```
5     int val = 100;
6     printf("val %d, sizeof %u, at address: %p \n", val, sizeof(val), &val);
7
8     int* pval = &val;
9     printf("pval %x, sizeof %u, at address: %p \n", pval, sizeof(pval), &pval);
10
11    *pval = 200;
12    *pval ++;
13    printf("final val: %d \n",
14    printf("final pval: %x \n",
```

Highest

Precedence	Operator	Description	Associativity
1	++ --	Suffix/postfix increment and decrement	Left-to-right
	()	Function call	
	[]	Array subscripting	
	.	Structure and union member access	
	->	Structure and union member access through pointer	
	(*type*){*list*}	Compound literal(C99)	
2	++ --	Prefix increment and decrement[note 1]	Right-to-left
	+ -	Unary plus and minus	
	! ~	Logical NOT and bitwise NOT	
	(*type*)	Cast	
	*	Indirection (dereference)	
	&	Address-of	
	sizeof	Size-of[note 2]	

Pointer: interesting case

```
5     int val = 100;
6     printf("val %d, sizeof %u, at address: %p \n", val, sizeof(val), &val);
7
8     unsigned char* pchar = (unsigned char*) (&val);
9     printf("pchar %x, point to value: 0x%02x / %u \n", pchar, *pchar, *pchar);
10
```

打印格式：%02x -- 2位宽的16进制数

NOTE, 指针类型转换

- &val 得出：int*
- 地址值不变，取地址的变量类型改变

```
val 100, sizeof 4, at address: 000000000061FE14
pchar 61fe14, point to value:  ??
```

Outline

SUSTech Southern University of Science and Technology

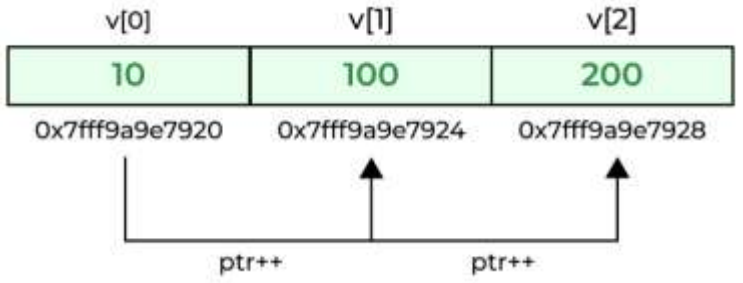
- Review
- Pointer: basic intro & interesting case
- **Pointer & Array**
- Assignment

Pointer & Array

```
12      char str[] = "hello world!";
13      // char* pchar = &str; // same as below
14      char* pchar = str;  // 指针类型与数组元素类型匹配
15      int str_len = 0;
16      while (*pchar != '\0')
17      {
18          printf("index %d: %c / 0x%02x / %d \n",
19                  str_len, *pchar, *pchar, *pchar);
20          str_len++;
21          pchar++;
22      }
23      printf("str_len %d \n", str_len);
```

```
index 0: h / 0x68 / 104
index 1: e / 0x65 / 101
index 2: l / 0x6c / 108
index 3: l / 0x6c / 108
index 4: o / 0x6f / 111
index 5:   / 0x20 / 32
index 6: w / 0x77 / 119
index 7: o / 0x6f / 111
index 8: r / 0x72 / 114
index 9: l / 0x6c / 108
index 10: d / 0x64 / 100
index 11: ! / 0x21 / 33
str_len 12
```

通过指针进行数组遍历



More,

- str_len 的计算过程，等同于标准库中strlen(...) 函数
- 数组的名称 （i.e. str），对应一个 constant pointer (常量指针)

Pointer & Array

SUSTech Southern University of Science and Technology

```
25      int vals[] = {0, 1, 2, 3, 4};
26      // int* pval_start = vals;
27      // int* pval_end = &vals[4];
28      vector_pointwise_square(vals, &vals[4]);
29      printf("%d %d %d %d %d",
30          vals[0], vals[1], vals[2], vals[3], vals[4]);
```

Highlight

- 通过指针减少值的传递
- 通过指针对数组元素修改

```
0 1 4 9 16
```

```
3   void vector_pointwise_square(int* pval_start, int* pval_end)
4   {
5       for (int* p = pval_start; p <= pval_end; p++) {
6           *p = (*p) * (*p);
7       }
8   }
```

Interesting for-loop

Outline

SUSTech Southern University of Science and Technology

- Review
- Pointer: basic intro & interesting case
- Pointer & Array
- **Assignment**

Assignment 1) Byte Encoder

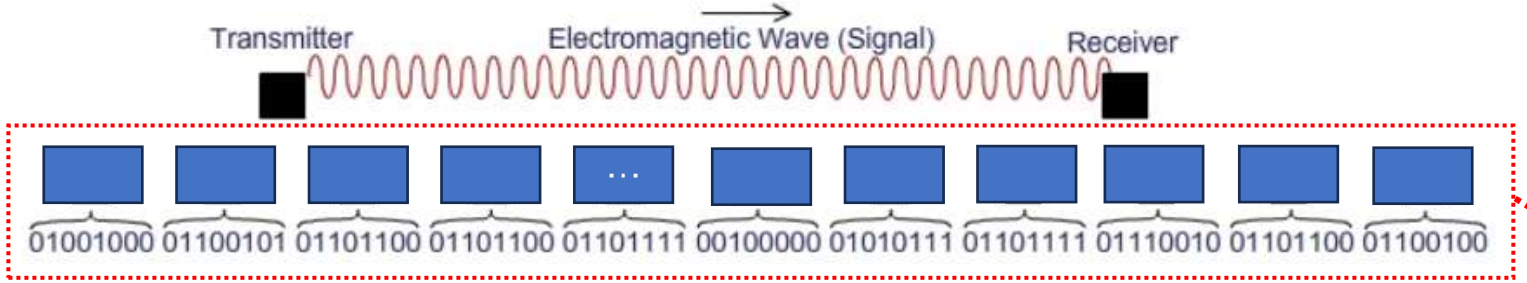
Write a program that accepts the following inputs:

- Student ID: A 64-bit unsigned integer representing a unique identifier for a student.
- Age: An 8-bit unsigned integer indicating the student's age.
- Height: A 32-bit floating-point number denoting the student's height in meters.

Task:

- Efficiently encode the above three pieces of information – the student ID, age, and height – into a compact byte array format (as below), ensuring the integrity of the original data when decoding.
- Print out each individual byte from the encoded byte array in hexadecimal format (with 0x as prefix).

```
00000001 18 1.75
0x01 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x12 0x00 0x00 0xe0 0x3f
```



Bytes Array

Little endian

Assignment 1) Byte Encoder

SUSTech Southern University of Science and Technology

Input

The first line contains an integer N ($1 \leq N \leq 10000$).

For the following N lines, each line contains an unsigned 64-bit integer, an unsigned 8-bit integer, and a 32-bit float number.

Hint

```
scanf("%llu", &student_id);
```

Output

Print each byte in the encoded byte stream. Each byte should start with `0x`. **Note that all alphabet character including the `x` should be lowercase.

One more testcase ☺

```
5
16842398300944799002 13 1.13762
16807921689683419112 6 1.60632
15773139836139754950 70 1.57582
8951828593339113904 30 1.90305
9002599733431277819 48 1.71120
```

```
0x1a 0x15 0x75 0x15 0x05 0x38 0xbc 0xe9 0x0d 0x88 0x9d 0x91 0x3f
0xe8 0xdf 0x29 0x7e 0xb9 0xbb 0x41 0xe9 0x06 0xe5 0x9b 0xcd 0x3f
0xc6 0x55 0xac 0x7d 0x1c 0x73 0xe5 0xda 0x46 0x78 0xb4 0xc9 0x3f
0xb0 0xa1 0x29 0x44 0xad 0x48 0x3b 0x7c 0x1e 0x24 0x97 0xf3 0x3f
0xfb 0x98 0x32 0x7e 0xc2 0xa8 0xef 0x7c 0x30 0x9a 0x08 0xdb 0x3f
```

Note: all output in 1 line

Assignment 2) 字符删除

write a function that removes all specific characters from a string in place (without using additional memory).

Hint:

- You will need to write the function: void remove_char(char* str, char c)
  where str is a pointer to a NULL-terminated string and c is the character that needs to be removed from the string.
- You need to make modifications directly to the original string, not create a new copy of the string.
- Make sure the final string is NULL terminated.
- Consider using pointer arithmetic instead of array indexing to traverse and modify strings.
- Consider edge cases such as empty strings or strings that do not contain specific characters.

NOTE: Code template is given in next page!

```
hello world!
o
hell wrld!
```

Assignment 2) 字符删除

SUSTech Southern University of Science and Technology

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

#define MAX_LEN 1000001

void remove_char(char* str, char c)
{
    // TODO - add your code here
}

int main()
{
    char str[MAX_LEN] = {0};
    gets(str);

    char c;
    scanf("%c", &c);

    remove_char(str, c);
    puts(str);
}
```

Debug in local

```
hello world!
o
hell wrld!
```

A sample solution will be like this. And you will **only** need to submit this code snippet.

```c
#include "lib.h"

/* other includes */

void remove_char(char *str, char c)
{
    // TODO: finish this function
}
```

Commit in OJ

SUSTech Southern University of Science and Technology

THANK YOU