四川大学期末考试试题（闭卷）

（2018--2019 学年第 1 学期） B 卷

课程号：311046040 课程名称：系统级编程 任课教师：

适用专业年级：软件工程 2016 级 学号： 姓名：

考生承诺

我已认真阅读并知晓《四川大学考场规则》和《四川大学本科学生考试违纪作弊处分规定（修订）》，郑重承诺：

1、已按要求将考试禁止携带的文具用品或与考试有关的物品放置在指定地点；

2、不带手机进入考场；

3、考试期间遵守以上两项规定，若有违规行为，同意按照有关条款接受处理。

考生签名：

题　号	一(40%)	二(10%)	三(5%)	四(15%)	五(12%)	六(8%)	七(10%)
得　分							
卷面总分			阅卷时间				

注意事项： 1. 请务必将本人所在学院、姓名、学号、任课教师姓名等信息准确填写在试题纸和添卷纸上；

2. **请将答案全部填写在本试题纸上；**

3. 考试结束，请将试题纸、添卷纸和草稿纸一并交给监考老师。

评阅教师	得分

一、单项选择题（本大题共 20 小题，每小题 2 分，共 40 分）

提示：在每小题列出的四个备选项中只有一个是符合题目要求的，请将其代码填写在下表中。错选、多选或未选均无分

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20

1. Which of the following numerical operations is most likely to lead to loss of precision?
   a. Integer multiplication
   b. Integer addition
   c. Floating-point addition
   d. Floating-point multiplication

2. Which is the value of expression 0x23^0x64?
   a. 0x66
   b. 0x0
   c. 0x67
   d. 0x20

3. Given the array int md[3][4], which code segment will cause compiler error?

   a. for (int i = 0; i < 3; i++)
   ```
   for (int j = 0; j < 4; j++)
           *(md + i*4 + j) = 0;
   ```
   b. for (int i = 0; i < 3; i++)
   ```
   for (int j = 0; j < 4; j++)
           *(*(md+ i) + j) = 0;
   ```
   c. for (int i = 0; i < 3; i++)
   ```
   for (int j = 0; j < 4; j++)
           md[i][j] = 0;
   ```
   d. for (int i = 0; i < 3; i++)
   ```
   for (int j = 0; j < 4; j++)
           *((int*)(md) + i*4 + j) = 0;
   ```

4. Which statement has NO effect on stack?

   a. ADD

   b. JMP

   c. PUSH

   d. RET

5. Given the following code segment, which statement is right?

   Char str[10]="Hello";   int x = size of(str); int y = strlen(str);

   a. x is 5 and y is 5;

   b. x is 10 and y is 10;

   c. x is 10 and y is 5;

   d. x is 5 and y is 10;

6. Consider the following segment of a C program.
   ```
   int i = 99;
   int a[100];
   i = a[i + 1];
   ```
   Which of the following is true of the segment?

   a. I will have the value of the last element of the array a at the end of any execution of the segment.

   b. When executed, the program will be prematurely terminated by the operating system because of an illegal memory access.

   c. I will have the value 99 at the end of any execution of the segment.

   d. Execution will fail because a has the wrong size.

7. Why is the mantissa of a floating point number shifted left as far as possible?
   a. to retain as much precision as possible
   b. to avoid overflow
   c. to avoid underflow
   d. to align bit positions, simplifying addition

8. How does x86 assembly store the return value when a function is finished?
   a. The ret instruction stores it in a special retval register.
   b. By convention, it is always in eax
   c. It is stored on the stack just above the ebp of the callee
   d. It is stored on the stack just above all the arguments to the function

9. Immediately after the CPU executes an instruction that is neither a branch nor a jump instruction, the program counter
   a. remains unchanged
   b. is incremented by one
   c. is incremented to point to the following instruction
   d. has a value that cannot be determined without further information

10. Which of the following features apply to standard heap allocation in C?
    i) The size of heap objects must be known at compile time.
    ii) Heap memory must be explicitly allocated.
    iii) Heap memory is deal located when a function returns.
    b. II only.
    c. I and II only.
    d. I and III.
    e. I only.

11. Why is it wrong to return the address of a local variable?
    a. It allows illegal access to the variable from arbitrary functions.
    b. The local variable may be in a machine register.
    c. It is faster to return the value of the variable.
    d. The variable address is invalid after the return.

12. In C, which of the following is the best way to detect when a pointer is freed twice?
    a. Set pointers to NULL after freeing them.
    b. Flag all blocks as free or not, and check the flag when calling free ().
    c. Keep a log of addresses that have been freed and scan the log before calling free ().
    d. Modify free () to set the freed data to zero.

13. A memory pool is a large block of memory from which small objects are allocated piecemeal by breaking them off from the pool as required. Under which of the following conditions would such a scheme result in greatly improved performance?

    i) All objects allocated from the pool are freed at around the same time.

    ii) All objects allocated from the pool are of similar sizes.

    iii) A garbage collector takes care of freeing memory.

    a. I only.

    b. II only.

    c. III only.

    d. I and II only.

14. Two code fragments are i) and ii) as following:

    i)

```
M= 100000; N=100000;
int data[M][N];
For (j = 0 ; j < N; j++) {
   For (i = 0; i < M; i++) {
        Sum += data[ i ] [ j ];
   }
}
```

    ii)

```
M= 100000; N=100000;
int data[M][N];
For (i = 0; i < M; i++) {
   For (j = 0 ; j < N; j++) {
        Sum += data[ i ] [ j ];
   }
}
```

Which one is more efficient (better performance) if you program with Mat lab which is column-major order for matrix?

    a. Both are very efficient.

    b. i)

    c. ii)

    d. Both are not efficient.

15. A garbage collector

    a. Frees memory blocks that cannot be reached by dereferencing pointers.

    b. Frees all memory blocks that will not be accessed in the future.

c. Frees memory blocks marked as "delete able".

d. Removes old versions of local variables from the stack.

16. What is tsc?

a. a timer mechanism of OS

b. a timer mechanism of x86 platform, which is the short name of time stamp counter

c. a system call of OS

d. a timer mechanism of c library

17. Which of the following is not optimization technique?

a. code motion

b. loop unrolling

c. constant folding

d. memory aliasing

18. Which of the following manages the transfer of data between the cache and main memory?

a. Registry.

b. Operating System.

c. Hardware.

d. Compiler.

19. What can Linker do?

i) Symbol Resolution

ii) Resolution

iii) load or map the Executable object file from the disk to memory

a. i) and ii) only.

b. i) and iii) only.

c. i), ii) and iii).

d. iii) only.

20. In IA32 or X86, which exception returns back to the point where exception happens?

a. interrupt

b. trap

c. fault

d. Abort

评阅教师	得分

二、 bit operation（本大题共 2 小题，各 5 分，共 10 分）

Now complete the following functions according to the following rules.

Each "Expr" is an expression using ONLY the following:

1. Integer constants 0 through 0xFFFFFFFF inclusive.
2. Function arguments and local variables (no global variables).
3. Some of the problems restrict the set of allowed operators.

You are expressly forbidden to:

1. Use any control constructs such as if, do, while, for, switch, etc.
2. Define or use any macros.
3. Define any additional functions in this file.
4. Call any functions.
5. Use any other operations, such as &&, ||, -, ?, or [] :
6. Use any form of casting.

You may assume that your machine:

1. Uses 2s complement, 32-bit representations of integers.
2. Performs right shifts arithmetically.
3. Has unpredictable behavior when shifting an integer by more than the word size.

```
/* least Bit Pos - return a mask that marks the position of the
 *                     least significant 1 bit. If x == 0, return 0
 *   example: least Bit Pos (96) = 0x20
 *   legal ops:! ~ & ^ | + << >>
 *   max ops: 6
 *   rating: 4
 */
int least Bit Pos(int x) {



   }
```

```
/*
 * logical Shift - shift x to the right by n, using a logical shift
 *   Can assume that 1 <= n <= 31
 *     Examples: logical Shift(0x87654321,4) = 0x08765432
 *     Legal ops: ~ & ^ | + << >>
 *     Max ops: 16
 *     Note: data type int indicates it is mathematical shift
 */
int logical Shift(int x, int n) {





}
```

评阅教师	得分

三、memory allocation（本大题共 1 小题，共 5 分）

Are there any errors in the following program? If there are, please correct them on the right side.

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

void get memory(char *p)
{
        p=(char *) malloc(100);
        strcpy(p,”hello world”);
}

int main( )
{
  char *str=NULL;
  get memory(str);
  printf(“%s/n”,str);
  free(str);
  return 0;
}
```

评阅教师	得分

四、stack discipline（本题共 15 分）

Stack discipline. Consider the following C code and its corresponding 32-bit x86 machine code. Please complete the stack diagram on the following page.

```
int bar (int a, int b) {
        return a + b;
}

int foo(int n, int m, int c) {
        c += bar(m, n);
        return c;
}
```

```
 int bar (int a, int b) {
00401030          push     ebp
00401031          mov      ebp,esp
00401033          sub      esp,40h
00401036          push     ebx
00401037          push     esi
00401038          push     edi
......
0040104B          add      eax,dword ptr [ebp+0Ch]
......
00401054          ret

 int foo(int n, int m, int c) {
00401060          push     ebp
00401061          mov      ebp,esp
00401063          sub      esp,40h
00401066          push     ebx
00401067          push     esi
00401068          push     edi
00401069          lea      edi,[ebp-40h]
0040106C          mov      ecx,10h
00401071          mov      eax,0CCCCCCCCh
00401076          rep stos dword ptr [edi]
00401078          mov      eax,dword ptr [ebp+8]
0040107B          push     eax
0040107C          mov      ecx,dword ptr [ebp+0Ch]
0040107F          push     ecx
00401080          call     @ILT+0(_bar) (00401005)
00401085          add      esp,8
00401088          mov      edx,dword ptr [ebp+10h]
0040108B          add      edx,eax
0040108D          mov      dword ptr [ebp+10h],edx
00401090          mov      eax,dword ptr [ebp+10h]
00401093          pop      edi
......
004010A3 C3       ret
```

1. (12 points) Draw a detailed picture of the stack, starting with the caller invoking foo (3, 4, 5), and ending immediately before execution of the ret instruction in bar. Return address in function that called foo : 0x004010E3
   The diagram starts with the address arguments for foo().

Address no.	value	description
0xffffd850		
0xffffd84c		
0xffffd848		
0xffffd844		
0xffffd840		

Address no.	value	description
0xffffd83c		
0xffffd838		
0xffffd834		
0xffffd830		

2. (1.5 points) What is the final value of ebp, immediately before execution of the ret instruction in bar?

   ebp=0x____________________

3. (1.5 points) What is the final value of esp, immediately before execution of the ret instruction in bar?

   esp=0x____________________

评阅教师	得分

五、performance optimization（本大题共 2 小题，共 12 分）

1. (6 points) We are considering an enhancement to the processor of a web server. Then new CPU is 30 times faster on search queries than the old processor. The old processor is busy with search queries 80% of the time, what is the speedup gained by integrating the enhanced CPU?

2. (6 points) Read the following two program fragments. Do they have same behavior in any circumstances? Explain your answer.

```
void twiddle1 (int *xp, int *yp)   {
    *xp += *yp;
    *xp += *yp;
}

void twiddle2 (int *xp, int *yp)   {
    *xp +=2 *yp;
}
```

评阅教师	得分

六、cache（本大题共1小题，共8分）

Read the code as following:

```
int i
int sum=0;
int data[100000];
for (i = 0; i < 100000; i += stride)
        sum += data[i];
```

Assume that size of int is 4-byte and the cache line (block) length is 16-byte.

If the stride is 1,2,4,8, what is their hit ratio and miss ratio? Fill the blanks in the table and give your explain.

Stride	Hit ratio	Miss ratio
1		
2		
4		
8		

评阅教师	得分

七、Exception（本大题共 1 小题，共 10 分）

According to our textbook, what is the relationship between Exception and the OS? Please give your point on this, and drawing a figure is suggested.