1101 NTOUCSE Program Design 1C Final

姓名：_____ 系級：_____ 學號：_____ 111/01/18 (二)

考試時間：**13:20 – 16:00**

Exam rules：
1. **Close book**, **close** everything including quizs, homeworks, assignments, reference materials, etc.
2. You can answer the questions in **English** or in **Chinese**, in this **problem sheet**.
3. You can use language features not taught in class if you feel necessary, but strictly limited in C or C++.
4. **No** mobile phone, pad, computer or calculator is allowed. (Electronic) English dictionary is OK
4. No peeping around! No discussion! No exchange of any material; **raise your hand if you have any question about the exam problems**
5. If you turn in the paper earlier than the specified time, **leave the classroom immediately and quietly**
6. Against any of the above rules will be treated as cheating in the exam and handled by school regulations.
7. **Turn in the signed problem sheet**.

1. Please finish the program according to the following requirements. The program should read in two positive integers **a** and b, calculate the sum of all odd integers between **a** and **b** (including both ends)

(a) [5] Please finish the following for loop and selection statement

```
int sum, i, a, b;
scanf(_____);
for (sum=0,_____; _____; _____)
    if (_____) sum += i;
```

(b) [3] the **if** statement in (a) can be modified as sum += _____ ? _____ : _____ ;

(c) [3] Please use the initialization condition of the **for** statement to control the addition of suitable numbers (do not use selection statement as in (a) and (b) above)

```
for (sum=0,_____; _____; _____)
    sum += i;
```

(d) [3] Please use integer arithmetic characteristics to select all odd numbers in the summation

```
for (sum=0, i=1; i<=b; i++)
    sum += i * _____;
```

(e) [5] Please derive and implement the trapezoidal formula for the summation of all odd integers in the closed range [a,b]

sum = _____ ;

(f) [6] Please complete the design of the following recursive solution for the summation of all odd integers in the closed range [a,b]

```
int sum(int a, int b) {
    return a>b ? ____ : a+_____;
}
int main() {
    …define used variables and read from keyboard two arbitrary positive integers a and b …
    printf("%d\n", sum(_____, b)); // hint: the first odd integer greater or equal to a
    …
}
```

2. Please write a **recursive** program to find the 9-digit integer exhibiting the following characteristics
   the left most digit is divisible by 1
   the left most 2-digit number is divisible by 2
   the left most 3-digit number is divisible by 3
   ◦ ◦ ◦
   the complete 9-digit number is divisible by 9

   (a) [10] If each digit is an arbitrary member of the set {1, 2,3,4,5,6,7,8,9}, please finish the following: The parameter **num** of the function **find** is an (**ith**-1)-digit integer satisfying the first **ith**-1 requirements. This function tries to find the **ith** digit till the 9-th (right most) digit satisfying the **ith**, **ith**+1, …, and the 9-th requirements.

```
#include <stdio.h>
int find(int num, int ith) {
    int i, c;
    if (_____________________) { // terminating condition of the recursion
        printf("%d\n", num);
        return 1;
    }
    else
        for (i=_____; i<_______; i++) // try all possible digit in the range 1~9 as the ith digit
            if ((c=num*10+i)%________==0)
                if (find(_____________, ____________)==1)
                    return 1;
    return 0;
}
int main() {
    find(0,1);
    return 0;
}
```

   (b) [10] While program in part (a) finds the first 9-digit integer satisfying the requirement, please modify the program such that it prints all possible 9-digit integers, counting the number of them and print out the resulting count.

(c) [10] We can modify the above program in part (a) such that each digit in the 9-digit integer is a **different** digit in the set {1,2,3,4,5,6,7,8,9}. The **digits** array in the following program is designed to store these 9 numbers, each can be used only once. At the beginning, this array stores the set {1,2,3,4,5,6,7,8,9}. After one digit is used, the count of the array members is decreased by 1 and that integer is exchanged with the 9-**ith** member such that there are 10-**ith** integers to be chosen from to determine the **ith** digit.

```
#include <stdio.h>
void swap(_____, int i, int j) { // exchanging the i-th and the j-th element of the array
    int tmp = _____;
    _____ = _____;
    _____ = tmp;
}
int find(int num, int ith, int digits[]) {
    int i, c, r;
    // the same terminating condition as in part (a)
    for (i=0; i<_____; i++) // try out all remaining digits in the digits array for the ith digit
        if ((c=num*10+_____)%_____==0) {
            swap(digits, _____, _____);
            if (r=find(_____, _____, _____))
                return r;
            swap(digits, _____, _____);
        }
    return 0;
}
int main() {
    int digits[]={1,2,3,4,5,6,7,8,9};
    find(0,1,digits);
    return 0;
}
```

3. In a game program, a character possesses the 3-dimensional spatial position, velocity, and the mass

(a) [5] Please define a user-defined structure **body** with three fields to hold the attributes above, in which the position and the velocity are each three double precision floating numbers for holding the x, y, z dimensions, the mass is an integer

(b) [5] Assume that the variable **n** holds an integer within the range 500~1000, please define a pointer

army that points to a two-dimensional array with **n** rows and **10** columns of element, which is a **struct body** structure

(c) [10] Please allocate with malloc() a two dimensional array with **n** rows and **10** columns of element, which is a **struct body** structure. The result is kept in the pointer variable **army**. Please check if the allocation succeed. In case of failure, print out an error message and stop the program.

(d) [10] Please define a function **readData**, which is to be invoked by readData(n, army). In this function, the file "sprites.dat" is to be opened and the n*10 set of **struct body** data is to be read into the army array in a loop. Each **struct body** data contains 7 numbers in a line of the file. Close the file after all data is read in.

(e) [10] Please write the **comp()** function for the qsort. The loop below sort the data in each row of the army array according to the mass and the Euclidean distance to the origin (0,0,0), in particular the mass is in descending order and in case of equal mass, the distance is in ascending order

```
int comp(const void *a, const void *b) {




}
for (i=0; i<n; i++)
    qsort(_________, _________, _________, _________);
```

(f) [5] Please deallocate the memory after suitable check of the pointer value