國立中正大學106學年度碩士班招生考試試題

系所別：資訊工程學系-甲組　　　　科目：軟體設計

1. (5%) Which of the following C statements will generate a random number within the integral range: 10, 11, …, 14, 15?
   a. (rand( ) % 15)+11;
   b. (rand( ) % 16)+10;
   c. (rand( ) % 5) + 11;
   d. (rand( ) % 6) + 10;
   e. None of the above statements.

2. (5%) Write down the output of the following C program

```
#include <stdio.h>
int main()
{
    int i, count=0;
    for(i=0 ;i<33; i++){
        if(i%3==0) continue;
        count++;
    }
    printf("%d\n", count);
    return 0;
}
```

3. (5%) Write down the output of the following C program

```
#include <stdio.h>
int x=1;
int func(int x){
  return x++;
}
int main(void){
  printf("%d\n", func(x));
  printf("%d\n", x);
  return 0;
}
```

4. (5%) Write down the output of the following C program

```
void main()
{
   int s[5]={5, 4, 3, 2, 1};
   int *p=s , *ptr=s+2 ;
   printf("A:%d\n", *p+2);
   printf("B:%d\n", *ptr);
   printf("C:%d\n", s[0]);
   printf("D:%d\n", *p++);
   printf("E:%d\n", *p);
}
```

5. (5%) Write down the output of the following C program

```
#define A  1
#define B  2
#define C  3
int hanoi(int N, int from, int to, int using)
{
   static int count=0;
   if (N > 0) {
      hanoi(N-1, from, using, to);
      count++;
      hanoi(N-1, using, to, from);
   }
   return count;
}
int main (void)
{
   printf(“%d\n”, hanoi(5, A, C, B));
   return 0;
```

6. (5%) We want to write a C program that exchanges the values of two integers. The main body is given below:

```
#include <stdio.h>
int main(int argc, char **argv) {
   int a, b;
   scanf("%d", &a);
   scanf("%d", &b);
   swap(&a, &b);
   printf( "%d\t%d\n", a, b);
   return 0; }
```

a. (2%) Write a short C function **swap** that fits into the program above, and exchange the values of the two variables **a** and **b**.

b. (3%) Now we rewrite the program in C++, in which case the main body is now:

```
#include <iostream>
using namespace std;
int main( int argc, char **argv) {
   int a, b;
   cin >> a >> b;
   swap(a, b);
   cout << a << "\t" << b << endl;
   return 0; }
```

Write a short C++ function **swap** that fulfills the same task as (a).

7. (5%) Let A = [a_{ij}], 1 <= i, j <= n be an n x n matrix, then its transpose B = A^t is also an n x n matrix that is defined by: B = [b_{ij}], 1 <= i, j <= n and $b_{ij} = a_{ji}$, for all i, j. Now we write a C++ style function to compute the transpose of an 10 x 10 matrix A, and store it in the same **double** array:

```
void transpose(double A[10][10]) {
   for (int i = 1; i < 10; i++) {
      for (int j = 0; j < i; j++) {
         // Fill the code here
      }
   }
   return; }
```

Fill the missing part in the program above. You should **not** introduce any new blocks.

8. (15%) Consider the following C++ code (recall that default constructors, i.e., constructors without arguments, do not need to be called explicitly in C++):

```
#include <iostream>
#include <cstdlib>
#include <cassert>
```

```
class A {
  public:
    A() { a = 1; }
    int a; };

 class B: public A {
   public:
     B() { a += 2; }};

 class C: public A {
   public:
     C() { a *= 3; }};

 class D: public B, public C { };

int main() {
  D *d = new D ();
  std::cout << d->a;
  return 0; }
```

a. (2%) The code is rejected by the compiler. Briefly explain why.
b. (4%) Solve the problem by changing only the class signatures, without removing classes from the inheritance declarations.
c. (4%) What is the output of running the program after your changes?
d. (5%) We now add the following **public** method to **A**:

```
virtual int get() { return a; }
```

and the following two classes:

```
class X : public A { };
class Y : public D, public X { };
```

and we write a **new** implementation of the method **main()**:

```
int main() {
  Y *y = new Y();
  B *b = y;
  b->a = rand(); // non-deterministic value assignment
  assert ( b->a == y->get() );
  return 0; }
```

Again, the code is rejected by the compiler. Solve the problem by adding code the body of class **Y**. Your solution should make the assertion valid.

9. (12%) Given a parameter k, a fixed-cost random number generator function RDG(k), and input size n, estimate the time complexity of the following function SEGMENT (n, k) by (a) writing its recurrence equation (including base case) (7%); and (b) solving it with Θ-notation (5%).

```
SEGMENT ($n$, $k$)
if $n < 1$
return NIL
for $i = 1$ to $n$
    do for $j = 1$ to $\sqrt[3]{n}$
        do RDG( $k$ )
if $n \geq 1$
  for $i = 1$ to 100
      length = $\lceil n/10 \rceil$
      SEGMENT ($length$, $k$)
```

10. (8%) Determine which of (1) Dijkstra's algorithm; (2) Depth first search; (3) greedy algorithm; (4) Bellman-Ford algorithm; (5) dynamic programming could be the best fit to solve the following problems: (a) Single-source shortest-paths problem with negative-weight edges (2%); (b) topological sort (2%); (c) fractional Knapsack problem (2%); (d) construct Huffman coding (2%).

11. (5%) To disprove (by providing counter example) or prove the argument: Given n distinct numbers, any comparison based sorting algorithm needs at least $nlogn$ comparisons in the worst case.

12. (10%) Briefly answer the following questions.
    a. What is the worst case time complexity of quicksort algorithm? (2pts)
    b. What is the average case time complexity of heapsort algorithm? (2pts)
    c. What is the best case time complexity of selection-sort algorithm? (2pts)
    d. What is the average case time complexity of searching in a sorted linked list? (2pts)
    e. What is the worst case time complexity of searching in a 2-3 tree? (2pts)

13. (15%) Please write a C program to generate a BST (Binary Search Tree) for the input keywords and use InorderTraversal to print out the keywords. The keywords can be read in from stdin by

    fgets(char *line, int MaxLine, stdin).

    For example, if the input data contains

    ```
    banana
    apple
    day
    candy
    ```

    The program will generate a BST for these keywords and then call InorderTraversal() to print out the keywords. The result for the example data will be

    ```
    apple
    banana
    candy
    day
    ```

    Hint: You need to declare the data structure for the BST node and your program may contain three functions: main(), BST_Insert(), and InorderTraversal().