國立清華大學命題紙

工程管理 一般

八十四學年度 工業工程 所 甲、丙 組碩士班研究生入學考試

科目 計算機概論 科號 2703 2903 2504 共 8 頁第 1 頁 *請在試卷【答案卷】內作答

壹. 解釋名詞 (請將下列電子計算機名詞之英文原文寫出，並簡要說明)： 12%

1. FORTRAN
2. BIT
3. I/O
4. DOS
5. CPU
6. ALU

貳. 問答及計算題：

1. 4% 請說明 FORTRAN 原始程式敘述中 1-80 欄位 (Column) 所代表的意義，並舉例說明。

2. 下列 FORTRAN 程式敘述執行後的結果如何？

(1) 4% 求 T 的值爲多少？

```
REAL  R, S, T
R = 4.0
S = 10.0
T = 1.0 / (1.0/S) + (1.0/R)
```

(2) 4% 求 Y 的值爲多少？

```
REAL  X1, X2, X3, Y
X1 = 5.0
X2 = 5.0
X3 = 0.5
 Y = X1 + X2/X1 - X2/X3
```

(3) 4% 下列DO LOOP 執行多少次？

```
DO 10 COUNT = -2, 14
```

(4) 3% 設 A = 5.5, B = 1.5, I = -3, DONE = .FALSE.
試問下列邏輯表示結果爲眞(true) 或爲假 (false)？

(1) -1.LE.I + 6
(2) (ABS(I).GT.3).OR.DONE
(3) A+B.GE.6.5

(5) 4% 設 $X = 11010101_2$ 求
(1) X 的 2'S complement (2 的補數)
(2) X 的 1'S complement (1 的補數)

(6) 4% 試問下列 FORTRAN 程式敘述，列印出的 IA 及 IB 值各為多少?

```
      IA = 0
      IB = 28
   10 IF(IA.GE.IB) GO TO 20
      IA = IA + 2
      IB = IB - IA
      GO TO 10
   20 WRITE (*,*) IA,IB
      .
      .
      .
```

(7) 11% 請回答下列問題:
(1) 何謂 Address Bus, Data Bus 及 Control Bus ?
(2) 何謂 General Purpose Register 及 Special Purpose Register?
(3) 何謂 Addressing Mode (定址模式)?

參. Given the following answer list,

(A) Small-talk (B) SQL (C) BASIC (D) FORTRAN (E) Objective-C (F) PROLOG (G) COMMON LISP (H) C (I) C++ (J) COBOL (K) PASCAL (L) PL/1

1. 3% Select the non-procedural language/s.

2. 3% Select the common use language/s for Artificial Intelligence programming.

3. 3% Select the language/s support object-oriented programming.

4. 3% Select one language most common use in relational database system.

肆. Computation problems:

1. 4% What is the maximal size of memory for a 32 bit computer?

2. 4% Use the symbol 0, 1, 2, 3, ..., 8, 9, A, B, C, D, ..., X, Y, Z, a, b, c, d, ..., x, y, z, totally, 62 characters to represent the number system with base 62. Show the result of the sum of $E8h_{62}$ and $2yB_{62}$,in number with base 62.

伍. Briefly answer the following data structure questions. Please do not write more than 80 words for each question.

1. 4% What is a hashing function and what is a good hashing function?

2. 4% What is a stack? What is a queue?

陸. Programming Problem:

1. 6% Please give more descriptive names of the following two C functions f1 and f2, and briefly explain what purpose they serve.

```
void f1(char *s, char *t)
{
        while (*s++ = *t++)
                ;
}
int f2(char *s, char *t)
{
        for ( ; *s == *t; s++, t++)
                if (*s =='\0')
                        return 0;
        return *s - *t;
}
```

2. 8% The content of file "capacity.txt" is:

```
0   100.0
1   120.0
2   125.0
3   150.0
4   50.0
5   250.0
6   240.0
7   310.0
8   210.0
9   60.0
```

The content of file "load.txt" is:

```
1  1  110.0
1  2  60.0
2  1  80.0
2  2  70.0
3  2  80.0
3  3  200.0
3  4  180.0
3  5  100.0
3  6  30.0
3  7  160.0
4  1  200.0
4  6  140.0
6  5  120.0
6  6  140.0
```

After executing the following C program by typing "a.out capacity.txt load.txt out.txt" under UNIX prompt, what is the content of out.txt?

```
#include <stdio.h>
float abc();
main(argc,argv)
int argc;
char *argv[];
{
        FILE *capacityF, *loadF, *outF, *fopen();
        float queue[10][10];
        float load[10][10];
        float cap[10];
        float temp;
        int mach, per;
        float loading, capacity;

        capacityF=fopen(argv[1],"r");
        loadF=fopen(argv[2],"r");
        outF=fopen(argv[3],"w");

        for (mach=0; mach < 10; mach++)
          {
            cap[mach] = 0.0;
            for (per=0; per < 10; per++)
              {
                queue[mach][per] = 0.0;
                load[mach][per] = 0.0;
              }
          }

        while( fscanf(capacityF,"%i %f", &mach ,&capacity)!=EOF)
            cap[mach]=capacity;

        while( fscanf(loadF,"%i %i %f", &mach ,&per ,&loading)!=EOF)
            load[mach][per]=loading;
```

```
        for (mach=0; mach < 10; mach++)
          {
             temp = load[mach][0] - cap[mach];
             queue[mach][0]= abc( temp, 0.0 );
            for (per=1; per < 10; per++)
              {
                 temp = queue[mach][per-1] +
                        load[mach][per] - cap[mach];
                 queue[mach][per] = abc( temp, 0.0);
              }
          }

          fprintf(outF,"%f\n",  queue[3][5]);
          fprintf(outF,"%f\n",  queue[3][6]);
          fprintf(outF,"%f\n",  queue[3][7]);
          fprintf(outF,"%f\n",  queue[6][6]);

         fclose(capacityF);
        fclose(loadF);
        fclose(outF);
}

float  abc(a1,  a2)
float  a1;
float  a2;
      {
        return (a1 > a2) ? a1 : a2;
      }
```

3. 8% The following Dynamic Programming problem example is copied from the popular Operation Research Textbook "Introduction to Operations Research" by F.S Hillier and G. J. Lieberman. Please write a "recursive" C program to solve this problem, instead of using the Dynamic Programming solution procedure.

Example 2—Distributing Medical Teams to Countries

The WORLD HEALTH COUNCIL is devoted to improving health care in the underdeveloped countries of the world. It now has five *medical teams* available to allocate among three such countries to improve their medical care, health education, and training programs. Therefore, the council needs to determine how many teams (if any) to allocate to each of these countries to maximize the total effectiveness of the five teams. The teams must be kept intact, so the number allocated to each country must be integer.

The measure of performance being used is *additional person-years of life*. (For a particular country, this measure equals the country's *increased life expectancy* in years times its population.) Table 11.1 gives the estimated additional person-years of life (in multiples of 1,000) for each country for each possible allocation of medical teams.

Which allocation maximizes the measure of performance?

FORMULATION: This problem requires making three *interrelated decisions*, namely, how many medical teams to allocate to each of the three countries. Therefore, even though there is no fixed sequence, these three countries can be considered as the three

Table 11.1 **Data for the World Health Council Problem**

No. of Medical Teams	*Thousands of Additional Person-Years of Life* — Country 1	2	3
0	0	0	0
1	45	20	50
2	70	45	70
3	90	75	80
4	105	110	100
5	120	150	130

stages in a dynamic programming formulation. The decision variables x_n ($n = 1, 2, 3$) would be the number of teams to allocate to stage (country) n.

The identification of the *states* may not be readily apparent. To determine the states, we ask questions such as the following. What is it that changes from one stage to the next? Given that the decisions have been made at the previous stages, how can the status of the situation at the current stage be described? What information about the current state of affairs is necessary to determine the optimal policy hereafter? On these bases, an appropriate choice for the "state of the system" is

s_n = number of medical teams still available for allocation to the remaining countries ($n, \ldots, 3$).

Thus, at stage 1 (country 1), where all three countries remain under consideration for allocations, $s_1 = 5$. However, at stage 2 or 3 (country 2 or 3), s_n is just 5 minus the number of teams allocated at preceding stages. With the dynamic programming procedure of solving backward stage by stage, when we are solving at stage 2 or 3, we shall not yet have solved for the allocations at the preceding stages. Therefore, we shall consider every possible state we could be in at stage 2 or 3, namely, $s_n = 0, 1, 2, 3, 4$, or 5.

Let $p_i(x_i)$ be the measure of performance from allocating x_i medical teams to country i, as given in Table 11.1. Thus the objective is to choose x_1, x_2, x_3 so as to

$$\text{Maximize} \quad \sum_{i=1}^{3} p_i(x_i),$$

subject to

$$\sum_{i=1}^{3} x_i = 5,$$

and the x_i are nonnegative integers.

Using the notation presented in Sec. 11.2, $f_n(s_n, x_n)$ is then

$$f_n(s_n, x_n) = p_n(x_n) + \text{maximum} \sum_{i=n+1}^{3} p_i(x_i),$$

where the maximum is taken over $x_{n+1}, \ldots, x_3$ such that

$$\sum_{i=n}^{3} x_i = s_n$$

and the x_i are nonnegative integers, for $n = 1, 2, 3$. In addition,

$$f_n^*(s_n) = \max_{x_n = 0, 1, \ldots, s_n} f_n(s_n, x_n)$$

Therefore,

$$f_n(s_n, x_n) = p_n(x_n) + f_{n+1}^*(s_n - x_n)$$

(with f_4^* defined to be *zero*). These basic relationships are summarized in Fig. 11.3.

Consequently, the *recursive relationship* relating the f_1^*, f_2^*, and f_3^* functions for this problem is

$$f_n^*(s_n) = \max_{x_n = 0, 1, \ldots, s_n} \{p_n(x_n) + f_{n+1}^*(s_n - x_n)\}, \qquad \text{for } n = 1, 2.$$

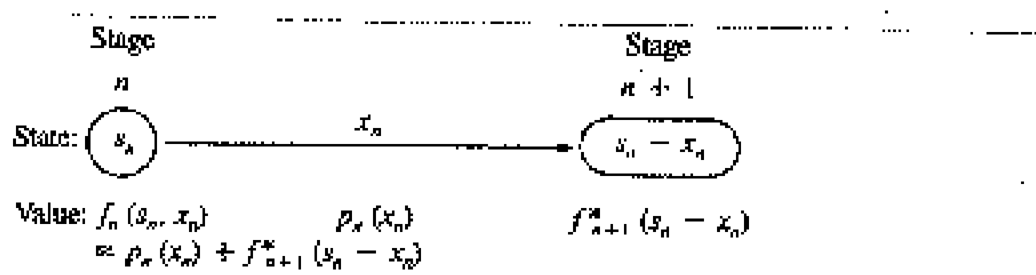

Figure 11.3 The basic structure for the World Health Council problem.

For the last stage ($n = 3$),

$$f_3^*(s_3) = \max_{x_3 = 0, 1, \ldots, s_3} p_3(x_3).$$