國立中央大學八十四學年度碩士班研究生入學試題卷

參考用

所別：電機工程研究所　甲組　科目：計算機概論　共 2頁 第 1 頁

1. Use a one-dimentional array and an interger variable to implement a Last-in First-out stack. Use the following declarations to write subroutines Pop() and Push(k). (25%)

```
--------------------------------------------------------------------------------
int Data[N], Head;                      /* data are all positive integer */

StackInit(){ Head = -1; }               /* initialize the stack */

int Push(k)                             /* return -1 if stack is full,
    int k;                                 else return 0 */
    { /* you need to write the subroutine here */}

int Pop()                               /* return -1 if stack is empty
                                           else the last pushed data */
    { /* you need to write the subroutine here */}
--------------------------------------------------------------------------------
```

Note: you can use either Pascal or C syntax to write the subroutines.

Example:

```
        Push(9)  -                return 0 (not full)
        Push(3)  -                return 0 (not full)
        Pop()    - return 3
        Push(7)  -                return 0 (not full)
        Pop()    - return 7
        Pop()    - return 9
        Pop()    - return -1 because the stack is empty
```

2. For a linked list, use both the following two styles to write programs to obtain the sum in the "data" field. (25%)

(a) Iterative style. Hint: use "while" or "for".

(b) Recursive style. Hint: recursive procedure calls.

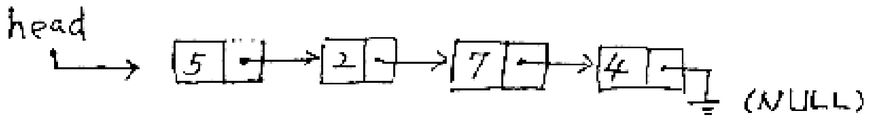
```
--------------------------------------------------------------------------------
typedef struct nodetype {
    int data;                     /* data that to be summed */
    struct nodetype *next;        /* point to the next list node */
}

struct nodetype *head;            /* point to the first node of the list */

int GetSum(x)                     /* return the summation of the "data"   */
struct nodetype  *x;              /* in the list */
{ /* you have to write the program here */}
--------------------------------------------------------------------------------
```

Note: you can use either Pascal or C syntax to write the subroutines.

Exp: For the following linked list, GetSum(head) return 18.



3. (16%)

(a) Represent decimal number 145 in binary, octal, hexadecimal, and BCD format respectively.

(b) use 6-bit 2's complement binary representation to represent 21, 7, -21, -7.

(c) Use the above reporesents to perform the following operations, 21+7, 21-7, 7-21.

(Detail the procedure in vertical form such as)

```
  3   0011
+ 2   0010
-----------
  5   0101
```

4. For the function $f = X\bar{Z} + X\bar{Y} + YZ$ (17%)

(a) Draw the Karnaugh map,

(b) Represent the function in sum-of-minterm form

(c) Represent the function in project-of-maxterm form

(d) Minimize the function using Karnaugh map (in sum-of-product form).

(e) Use AND, OR, and INV gates to implement the function.

(f) Use NAND gates only to implement the function.

5. Design a 3-bit down-count counter. (17%)

(a) Draw a state diagram (graph) of a 3-bit down-count counter.

(b) Draw the state table of (a).

(c) Use three D-type Flip-flops to design a 3-bit down-count counter based on the state table in (b).