

Exercise I

Draw the memory state of the following program:

```
#define N 5
typedef struct Node{
    int d;
    char *c;
} node;

void f(node *a, int i, char *pc){
    (a+i)->d=i;
    (a+i)->c=pc;
}

void main(){
    char *course="INFO216";
    int t[N]={1,2,3,4,5}, i;
    node tn[N];
    for(i=0;i<N;i++){
        f(tn,i,course);
    }
}
```

Exercise II

The C library function **strpbrk()** finds the first character in the string **str1** that matches any character specified in **str2**. This does not include the terminating null-characters. This function returns a pointer to the first character in str1 that matches one of the characters in str2, or NULL if no such character is found.

Following is the declaration for strpbrk() function: `char *strpbrk(const char *str1, const char *str2)`

The following example shows the usage of strpbrk() function:

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

int main () {
    char str1[] = "abcde2fghi3jk4l";
    char str2[] = "xhyf0b";
    char *ret = strpbrk(str1, str2);
    if(ret) {
        printf("The matching substring: %s\n", ret);
    }
    else {
        printf("None of %s characters can be found in %s", str2, str1);
    }
    return(0);
}
```

This will produce the following result: **The matching substring: bcde2fghi3jk4l**

You are asked to write your own strpbrk() function without using string.h functions (not even the strlen function).

Exercise III

An factory manages the orders of its customers using a linked list. An order is defined by the customer name, the ordered item code (string) and the quantity ordered.

1. Define in C the recursive structure representing an order.
2. Write a recursive function qtItem() which, given a list of orders and an item code, calculates and returns the total quantity ordered of this item.