

MC140.03
Final Exam
18 December 2000
2 hours 30 minutes

Name: _____

Please write all your answers on this exam paper.

If necessary, you may use the backs of pages, but indicate clearly that your answer continues on the back.

Please answer each question only once.

Please print clearly.

You may use one 8.5x11" page of notes for this exam.

Read through the entire exam before you answer any questions.

Then answer the questions in the order in which you feel most comfortable.

It is better to answer fewer questions properly than more questions sloppily.

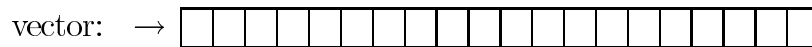
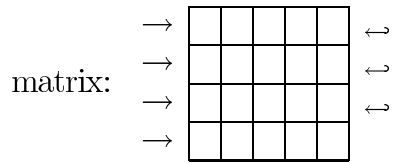
Good luck!

Have a great Christmas break!

problem	points	mark
1.	15 points	
2.	15 points	
3.	10 points	
4.	10 points	
5.	10 points	
6.	20 points	
7.	20 points	
TOTAL	100 points	

1. (15 points)

Write a function that transforms the contents of a matrix of integers (two-dimensional array) into a vector of integers (one-dimensional array), in the following order:



Your function should take the matrix, the vector, the number of rows in the matrix and the number of columns in the matrix. It should not return anything. You may assume that the vector is large enough to hold all of the entries from the matrix.

2. (15 points)

Write a function that takes a string as input and a specified character to look for in that string and returns the number of occurrences of that character in the input string. The search should not be case-sensitive (i.e., upper and lower case in the input string should both be counted). The search character will always be a capital letter, but you should search for it and its lowercase equivalent in the string.

For example, if the input is:

```
s = "The tendency of people to focus on the meaning of sentences  
    influences their ability to notice some of the obvious features."  
c = 'F'
```

then the output is:

```
output = 6
```

For reference, the ASCII value of 'A' is 65, the ASCII value of 'Z' is 90, the ASCII value of 'a' is 97, and the ASCII value of 'a' is 122.

3. (10 points)

What is printed by the following program?

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

int do_something(char s[], int i);

int main()
{
    char s[100] = "Life is like a box of chocolates.";

    printf("answer = %d\n",do_something(s, 0));
    return (0);
} /* end of main() */

int do_something(char s[], int i)
{
    if (i > strlen(s))
    {
        return (1);
    }
    else if (s[i] == ' ')
    {
        return (1 + do_something(s, i+1));
    }
    else
    {
        return (do_something(s, i+1));
    }
} /* end of do_something() */
```

4. (10 Points)

What do the following code fragments print?

Part a.

```
void what1(int x, int y)
{
    int j;

    for ( j = y; j > x; j-- )
        printf( "%d / %d ..\n", y, x + j );
} /* end of what1() */

int main (void)
{
    what1(3, 8);
    return (0);
} /* end of main() */
```

Part b.

```
int dooble(int x)
{
    return (x*2);
} /* end of dooble() */

void what2(int x, int y)
{
    if (x > y)
    {
        printf("%d\n", x);
        what2(x - 1, dooble(y));
    }
} /* end of what2() */

int main(void)
{
    what2( 20, 3 );
    return (0);
} /* end of main() */
```

5. (10 points)

What will be printed by the following program?

```
#include <stdio.h>

int foo(int *x, int *y);

int main(void)
{
    int a=3, b=4, c;

    c = foo(&a, &b);
    printf("%d %d %d\n", a, b, c);

    return(0);
} /* end of main() */

int foo(int *x, int *y)
{
    int i, j;

    *x = *x + *y;
    j = *x + *x;
    *y = j;
    i = *x;

    printf("%d %d %d %d\n", *x, *y, i, j);

    return(j);
} /* end of foo() */
```

6. (20 points)

Let x and y be integers and let (x, y) denote a *pair* of integers. A *set* of n pairs of integers, $\{(x_1, y_1), \dots, (x_n, y_n)\}$ can be represented in a C program by two arrays a and b where the i th element of array a has one element of a pair and the i th element of array b has the other element of the same pair.

Given the following definition to store a set of up to 100 such pairs with n storing the current number of pairs:

```
typedef struct
{
    int n;
    int a[100];
    int b[100];
} Pairset;
```

Part a.

Write a function:

```
int hasSelfPair(Pairset ps)
```

such that `hasSelfPair` returns 1 if `ps` contains a pair where its x element equals its y element. If `ps` contains no such pair, then `hasSelfPair()` should return 0.

Part b.

Write a function:

```
int findMax(Pairset ps)
```

which returns the index of the pair in **ps**, the sum of whose elements is the largest. The sum of a pair (x, y) is simply $x + y$. For example, if **ps** is the set of pairs $\{(2, 3), (3, 5), (3, 7)\}$, then **findMax(ps)** should return 2. If more than one pair has the same sum equal to the maximum sum, return the index of the first pair that has the maximum sum.

7. (20 points)

Part a.

Write the definition of a structure called `car` that contains a string for storing the make of the car (maximum length of 30 characters), a string for storing the model of the car (maximum length of 30 characters), an integer for storing the year the car was made and a string for storing the color of the car (maximum length of 20 characters).

Part b.

Write the definition of an array consisting of 100 `car` structures. You do not need to initialize any values for this array.

Part c.

Write a function called `printCar` that will take as input a pointer to a single `car` structure and will print out its contents.

Part d.

Write a function called `howMany` that takes the array of `car` structures, an integer containing the number of items in the array and a string that contains the name of a particular make of car. The function should return the number of cars in the array that are of the same make as the input string.