

UNIVERSITY OF TORONTO
FACULTY OF APPLIED SCIENCE AND ENGINEERING

FINAL EXAMINATION, APRIL 2001
APS 107 - COMPUTER FUNDAMENTALS

Examiners: D.C.S. Kuhn, M. Paraschivoiu and L. Shu

Student Name (print): _____

Student Name (signature): _____

Student Number: _____

DO ALL WORK IN THIS BOOKLET

IMPORTANT NOTE: If your C code program contains errors, part marks will be awarded for a correct algorithm. However, a statement of your algorithm is not required for full marks. Note also that marks will be awarded for correctness of your algorithm, C syntax, adherence to recommended C coding practice, code efficiency, and the clarity of your program.

Specified aids allowed (Type D).

All calculators allowed; no computers allowed.

Time allotted: 2 hours and 30 minutes

Question	Maximum Mark	Actual Mark
1	3	
2	3	
3	4	
4	4	
5	3	
6	3	
7	8	
8	10	
9	12	
Total	50	

Question 1. (3 marks) Sum 12 and -6 using binary arithmetic. Convert 12 and -6 to binary form, sum the numbers in binary form, and then convert the sum to decimal form. Assume the binary numbers are integers and are stored in 1 byte of memory.

Question 2: (3 marks) Consider the following code fragment:

```
int i;
char names[] = "Liam";
char *p = names;

for (i=0; i<4; i++)
    *(names+i) = *(names+3-i);
printf("%s\n", names+1);
printf("%d\n", sizeof(names));
printf("%p\n", p+2);
```

If p has value ABCD 0000, list in the box below the outputs of the above code fragment.

Question 3. (4 marks) A computer program is designed to read a string entered by the user, and make a copy of this string.

The program is shown below but contains at least 4 errors. In the table below, identify 4 and only 4 errors and their type (compile-time, run-time, or logic). Identify the original source of the error as well the first place where the program will likely experience compilation difficulties or run-time crashes, if they occur in different places. One line may have multiple errors. One error may involve multiple lines.

When appropriate, reference the line number where the error occurs. The line numbers on the left of the page are NOT part of the C program; they are included only for identification purposes.

Question 3. cont'd

Program Code

Line #

```

1      #include <stdio.h>
2      #define SIZE 10
3      void copy(const char *original, const char *copy);
4      main(){
5          char string1[SIZE];
6          char string2[5];
7
8          printf("enter a string:");
9          gets(string1);
10         copy(string1, string2);
11         return 0;
12     }
13     void copy(const char *original, const char *copy) {
14         int i;
15         for (i=0; i<SIZE; i++) {
16             copy+i = original[i];
17         }
18         return 0;
19     }
20

```

The 4 errors are:

Line #(s)	Error description	Error type

Question 4. (4 marks) What is the output for the following code if the address of the first element of the array is `ffffffc2a`?

```
#include <stdio.h>

main() {

    char array[10] = "012345678";

    char *ptr1 = &array[1];
    char *ptr2 = array;

    printf(" *(ptr1+2) = %d\n", *(ptr1+2));
    printf(" *(ptr1++) = %c\n", *(ptr1++));
    printf(" *(ptr1+1) = %c\n", *(ptr1+1));
    printf(" ptr2+1 = %p \n", ptr2+1);
}
```

Question 5. (3 marks) Below is a short working program. Using the box provided, write down the output produced by the program.

```
main()
{
    int *ptr, i, j;

    for (i=0; i<3; i++){
        ptr = malloc((i+1)*sizeof(int));
        for (j=0; j<i+1; j++){
            *(ptr+j) = 0;
        }
        *ptr=i+1;
        printf("The value of ptr[%d] is: %d\n", i, ptr[i]);
        free(ptr);
    }
}
```

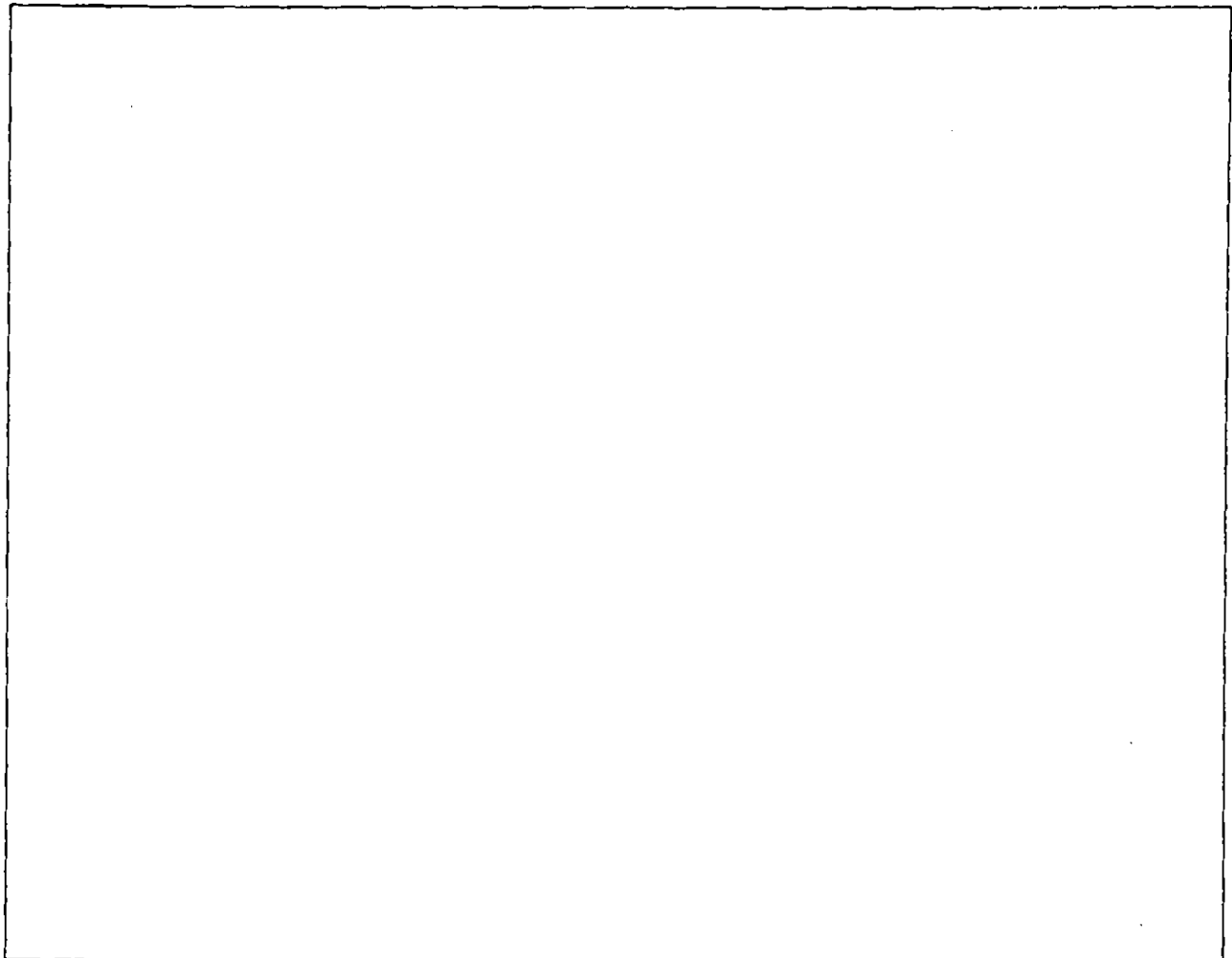
Question 6. (3 marks) Answer by true or false the questions below:

- a) If `arr` is the name of an array then `arr[i]` is the same as `*(arr+i)`. [TRUE or FALSE]
- b) To print using `printf` a percent sign you need to use `\%` in the literal string.
[TRUE or FALSE]
- c) It is a compile error to use the modulus operator (`%`) with anything other than integers.
[TRUE or FALSE]

Question 7. (8 marks) Write a C program to display a triangular shape made up of asterisk symbols (*). The user will input, from the keyboard the number of lines. Each line is made up of blank spaces followed by one or more asterisks. A sample of the output of your program when the user is inputting 3 is:

```
  *  
 ***  
*****
```

Note that the first line is made up of three blank spaces followed by an asterisk, the second line is made up of two blank spaces followed by three asterisks and the last line is one blank space and five asterisks.



Question 8, (10 marks) Write a function `square_a_string` that takes a string, converts it to its integer equivalent, and returns the square of the number. The function `square_a_string` is called from the main function as follows – write the function definition using pointer offset notation – do not use functions `atoi` or `atol`. An error flag is returned if the string contains characters that do not correspond to digits.

For example, if "12" is entered, the integer returned by the function is 144.

```
main(){
    char string[5];
    long answer;

    printf("enter string consisting of 4 or fewer digits ");
    scanf("%s", string);
    answer = square_a_string(string);

    if (answer != -1) printf("the square is %d\n", answer);
    else
        printf("string contained characters that do not correspond to digits");
    return 0;
}
```

Question 9: (12 marks) A rural parcel courier wants to optimize her delivery route. She has access to a file "parcel.dat" that contains the number of parcels to deliver, and for each parcel a parcel number and location. "parcel.dat" starts with the number of parcels to deliver, i.e. the header line. Following the header line for each parcel, the parcel number (8 binary digits with leading 1 digit) followed by its two coordinate delivery location defined in terms of kilometers east and north of the CN tower. A negative coordinate would indicate west and south respectively. A simple example of a data file is listed below:

```
Number of parcels: 5
11001100 23.75 -34.61
11101100 45.33 12.26
11010000 -54.00 134.62
11101000 -2.43 78.61
11000100 0.01 -12.54
```

In the above example parcel labeled 11001100 is to be delivered 23.75 km east and 34.61 km south of the CN tower.

Write a C program that reads the information from a parcel delivery file named "parcel.dat" and calculates the optimal route for the parcel courier to deliver the parcels. Assume delivery is by air and the parcels are dropped on the specified delivery location. To optimize her route she delivers the first parcel on the list, then the parcel that is located closest to the first parcel, then she delivers the parcel closest to the parcel just delivered etc. The program should print the parcel numbers in order of delivery including their positions. Assume there are never more than 63 parcels to deliver at one time. For the above example the output would be:

```
11001100 23.75 -34.61
11000100 0.01 -12.54
11101100 45.33 12.26
11101000 -2.43 78.61
11010000 -54.00 134.62
```

Recall the distance between two points in two-dimensional space is:

$$\text{distance} = [(x_1 - x_2)^2 + (y_1 - y_2)^2]^{1/2}$$

where x_1 and x_2 represents the east-west position of two delivery locations and y_1 and y_2 represents the north-south position of the two delivery locations.

Question 9 Cont'd.

