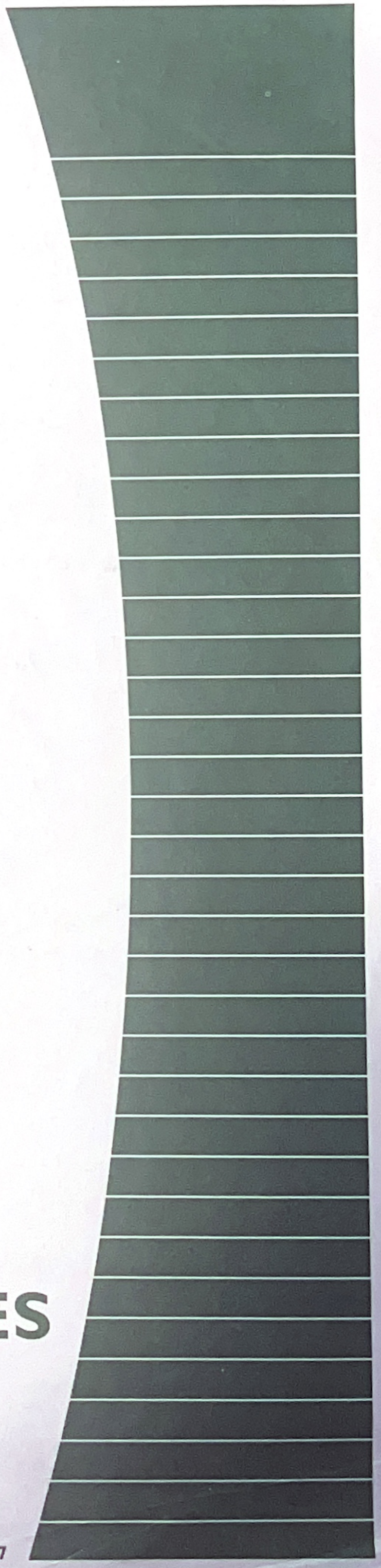
Hilroy



80 Pages
27.6 cm x 21.2 cm

Ruled 7 mm • Ligné 7 mm

EXERCISE BOOK
CAHIER D'EXERCICES

NAME/NOM Cole

SUBJECT/SUJET COMP 120

ASSEMBLED IN CANADA WITH IMPORTED MATERIALS
ASSEMBLÉ AU CANADA AVEC DES MATIÈRES IMPORTÉES

12107

Input/Output

[*needs <stdio.h>]

- scanf / reads information from the screen
- printf / writes information on the screen

- ex:

```
char grade = 'A'
```
*(single apostrophes for single characters)

```
printf("your grade in COMP 120 is %c", grade);
```
*(tells it to print a character) *(specifies which variable)

- \n (new line)
  \t (tab, *indent*)

- "CONTROL STRING" contains (text, *conversion specifier, escape sequence)

- CONVERSION SPECIFIERS:

Type	Specifier	
int	%d	
long int	%li	
float double	%f	(%e * puts it in scientific notation)
long double	%lf	(%le * scientific notation)
char	%c	

- %wi (specifies how wide before answer displayed)
  %w.df (specifies width and # of decimal places)

  Left-justify (ex: %-5i), moves it left

- ex: Suppose "num" has value 24.527810 (8 spots)

SPECIFIER	OUTPUT
%f	24.527810 (9 spots)
%12f	---24.527810 (12 spots)
%12.3f	-------24.528 (12 spots)(3 decimals)
%-12.3f	24.528 --------(12 spots)(3 decimals)
%+12.3f	---+24.528 (12 spots)(3 decimals)
%12.2f	--------24.53 (12 spots)(2 decimals)
%5f	24.527810 (because value is bigger) (would have to specify decimals)

- ex:

```
float mark;
char grade;
prinf("Enter a mark and grade: ");
scanf("%f, %c", &mark, &grade);
```

Week 2

Algorithms

Criteria for WELL-DEFINED:

Unambiguous (clear meaning)

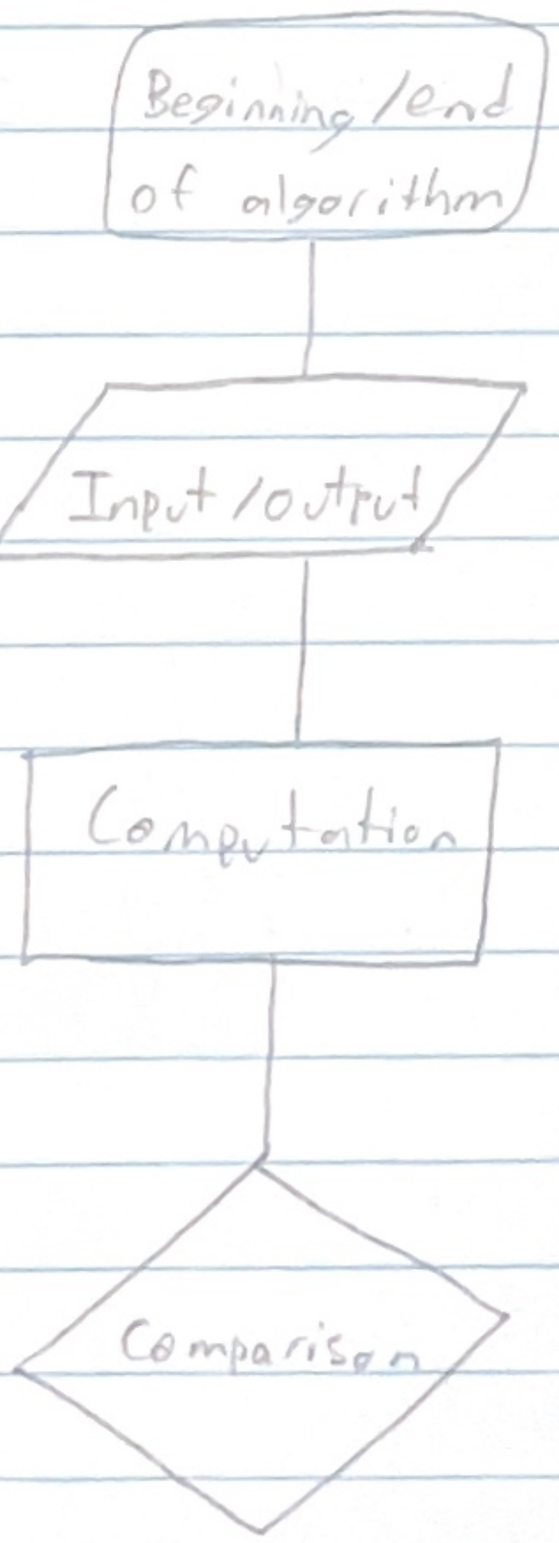
Deterministic (unique action is specified)

Feasible (can be done)

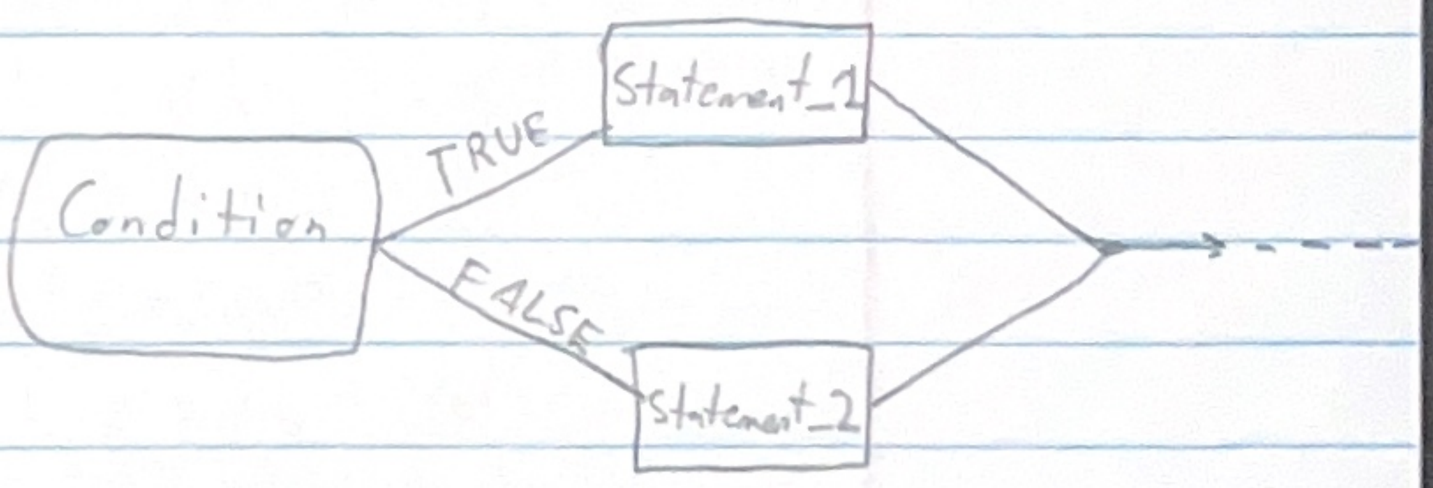
Finite (comes to an end)

"Pseudocode" / english-like statements (logical structure)

FLOWCHART:



"IF" Condition "THEN" statement_1
"ELSE" statement_2



Assignment 1

PSEUDOCODE:

① prompt user to enter binary number

② Assign decimal value ← 0

③ Get the first binary digit and Assign next digit ← value of the digit

④ WHILE (the nextDigit is 0 or 1)

(4.1) Assign decimal value ← (decimal value • 2) + nextDi

(4.2) Get the next binary digit and Assign nextDigit ← the value of the digit

⑤ Output the value of the variable decimal value

ex:

1 2 3 4 5

1 1 0 1 1

① dV = 0

nD = 1

② dV = 2 · dV + nD = 1

nD = 1

③ dV = 2 · 1 + 1 = 3

nD = 0

④ dV = 2 · 3 + 0 = 6

nD = 1

⑤ dV = 2 · 6 + 1 = 13

nD = 1

OUTPUT: dV = 2 · 13 + 1 = 27

nD = X

Boolean Expressions

True or False / Boolean expressions

0 represents false

all othe integers represent true

OPERATOR	INTERPRETATION
<	is less than
<=	is less than or equal to
>	is greater than
>>=	is greater or equal
==	is equal to
!=	is not equal to

LOGICAL OPERATOR	INTERPRETATION
!	not
&&	and
||	or

```
#include <stdlib.h>
#include <math.h>
#include<stdio.h>

int main()
[
    double a;
    double b;
    double c;
    double root_1;
    double root_2;

printf("\n Enter a, b, and c of equation\n");
Scanf("%lf%lf%lf", &a, &b, &c);

  if (b*b - 4*a*c>=0)
[
    root_1 = (-b + sqrt(b*b - 4*a*c)) / (2*a);
    root_2 = (-b - sqrt(b*b - 4*a*c)) / (2*a);

    printf("\nThe roots are: %f and %f. \n", root_1, root_2);

]
else
[
    printf("\nNO Real Roots!!! \n");

]
return 0;
```

LEC
Chp. 3

Functions

- Foo example:

i = 11, j = 5, k = 0

Function ①

k = ∅, 1
j = 11

Foo

Function ②

k = 0
j = 5
i = 11

main

- Local variables/contained in a function
- Global variables/contained outside the main fcu

- ex: 2

a = ~~x~~ 5
b = ~~3~~ 4
c = ~~10~~ 4

max2

a = -1
b = 3
c = 10

main

LEC
Chp. 3

Data Files

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

int main()
[
  File * inFile;
  inFile = fopen("myInputFile.txt", "r");

  if (inFile == NULL)
  [
      /* Print error message and exit */
  ]
  else
  [
      /* process file */
  ]
]
```

(reading *)

- **Reading from a file:**

```
int id;
double salary;

fscanf(infile, "%i %lf", &id, &salary);
```

- **Writing to a file:**

```
fprintf(outfile, "%i %lf\n", id, salary);
```

- **Closing a File:**

```
fclose(inFile);
fclose(outFile);
```

LEC

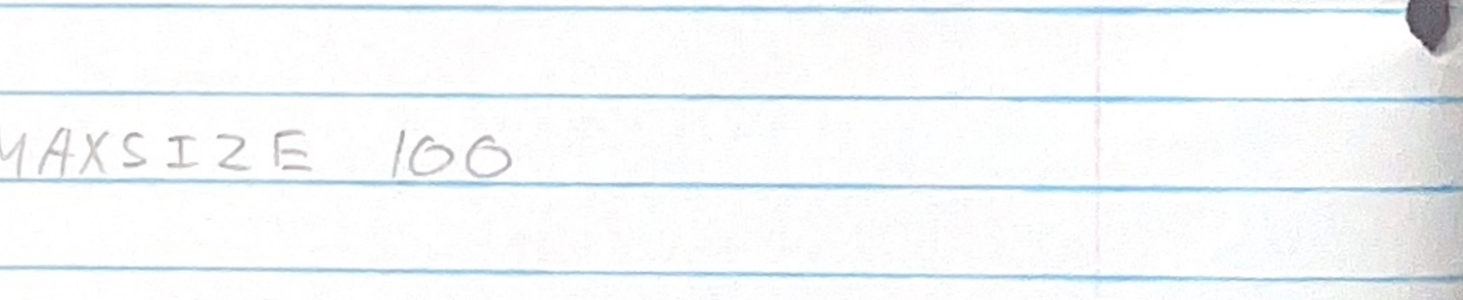
Arrays

- int a[10] declaring 10 int variables at the same time

- Array Data Structures (2 Types)
  - Random acess *(can be accesse d in any order)
  - Sequential access *(must be accessed in a specified

```
#define MAXSIZE 100
.....
int list[MAXSIZE];
```

- float data[4]




```
for (i=0; i < size; i++)
if (array[i] == wanted)
    return true;
```

```
#include <std.bool>
```

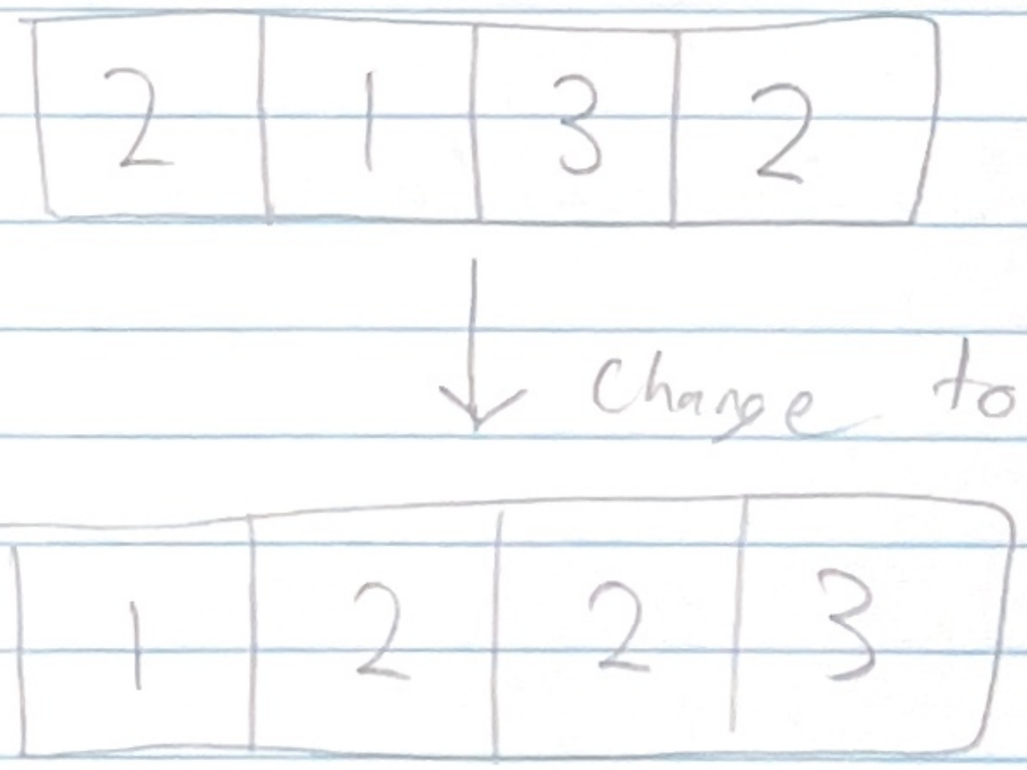
4 parameters

```
int CountDuplicates(char array1[], int size1, char Array2[],
                                                    int size2)
  int count= 0;
  int i;

  for (i=0; i<size1; i++)
```

rand %(max-min+1)+min

```
X = (float) rand() / RAND_MAX;
x = x*(max-min);
x = x + min;
```



LEC

Arrays

Swap Function

```
Void SwapElements (int a[], loc1, loc2)
{
    int tmp = a[loc1];
    a[loc1] = a[loc2];
    a[loc2] = tmp;
}

void SelectionSort (int a[], int size)
{
    int i;

    for (i=0; i < size-1; i++)
    {
        minIndex = findminIndex (a, size,
        if (a[minIndex] < a[i])
        {
            SwapElements (a, i, minIndex);
        }
    }
}
```

Big O Notation / Gives worst-case performance of an algorithm

```
int testArray(int numArray[], charletterArray[])
{
    int i;

for( i=0; i>SIZE; i++)
    {
        if(numArray[i]==letterArray[i])
            return -1;

    else return
```