Punched card

"Computer Science: An Overview" Pearson International Edition

Transistor

$1.(2^7+2^6+2^5+2^4+2^3+2^2+2^1+2^0)=255$

Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø

8 bit ⇄ 1 byte

2^8 symbols

- The two principal characteristics of a computer are: It responds to a specific set of instructions in a well-defined manner
- Dr. John Atanasoff and Clifford Berry invented first electronic computer.
- The first general-purpose computer in 1946 | J. Presper Eckert, John Mauchly Uses decimal data, Data Input with Punched Cards | ENIAC
- IBM Stretch Computer (Mainframe) 1959 (Banka)
  IBM 7030 was much slower than expected.
- Mainframe used primarily by corporate and governmental organization for critical applications, bulk data processing such as census, industry and consumer statistics, enterprise resource planning and transaction processing.

{ Second Generation / Transistors
  Third Generation / Integrated Circuits (Silicon chips, semiconductors)
  Forth Generation / Microprocessor

Von Neumann Architecture

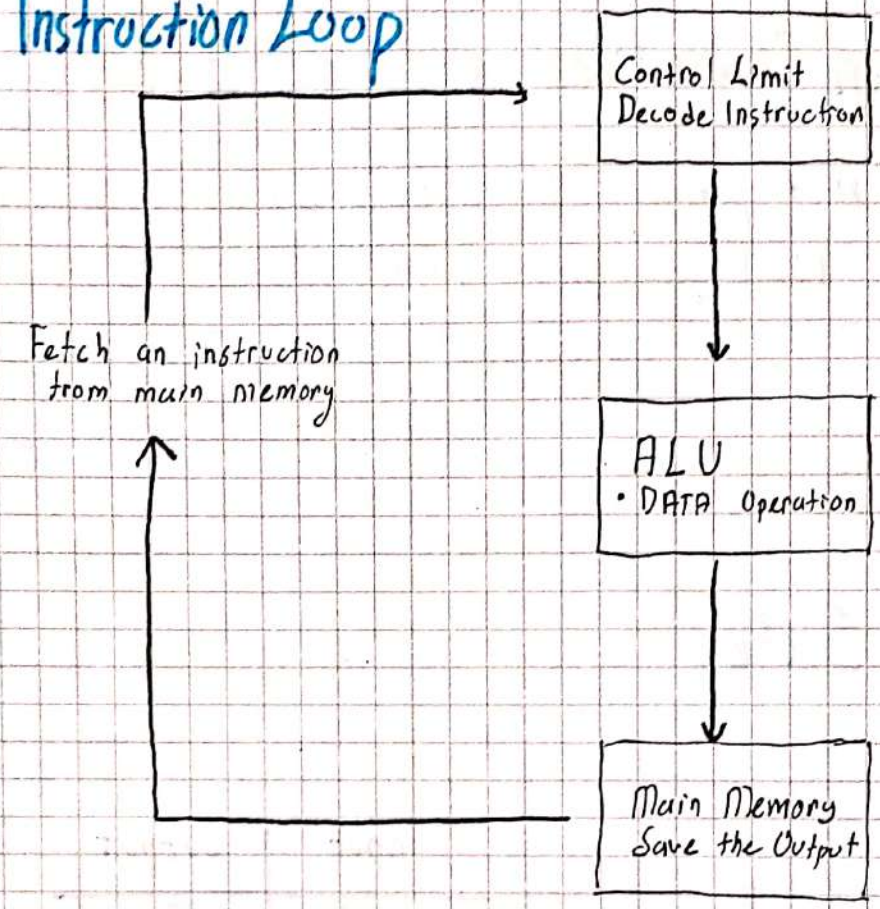
Binary data

Memory and arithmetic unit are managed by CPU

Kashin Color

Veri ve etkin kullanı

CPU Instruction Loop



Data Storage

Bit : Binary Digit → 0 / 1

8 bit → 1 bayt

1024 Byte → 1 KB

1024 KB → 1 MB

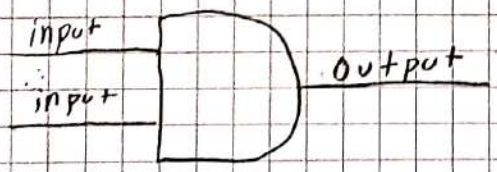
Boolean Operations and Logic Gates

A device that produces the output of a Boolean operation when given the operation's input values is called a GATE.

- AND
- OR
- XOR
- NOT

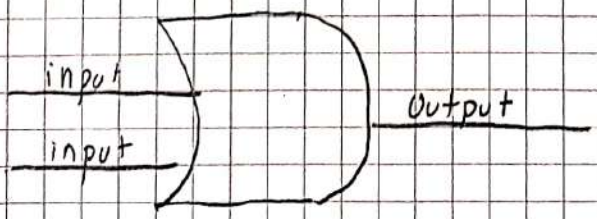
Logical Gates

AND



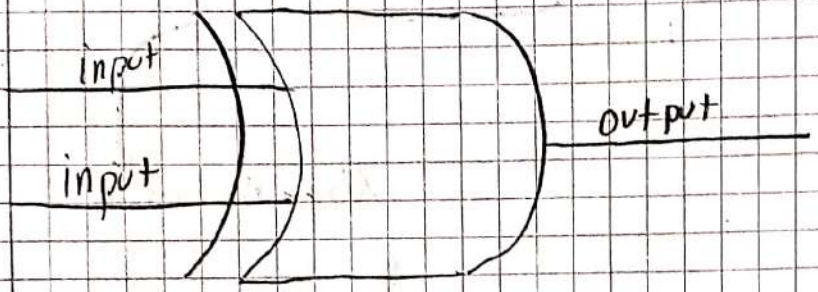
Input		Output
0	0	0
0	1	0
1	1	1
1	0	0

OR



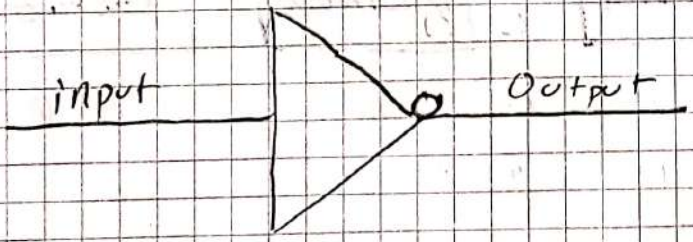
Input		Output
1	1	1
0	1	1
1	0	1
0	0	0

XOR



Input		Output
0	0	0
1	1	0
0	1	1
1	0	1

NOT



Input	Output
1	0
0	1

Number Systems

Positional

- Binary
- Octal
- Decimal
- Hexadecimal

Non-Positional

- Roman Number

Binary	Octal	Decimal	Hexadecimal
0000	0	0	0x0
0001	1	1	0x1
0010	2	2	0x2
0011	3	3	0x3
0100	4	4	0x4
0101	5	5	0x5
0110	6	6	0x6
0111	7	7	0x7
1000	$(10)_8$	8	0x8
1001	$(11)_8$	9	0x9
1010	$(12)_8$	$(10)_{10}$	0xA
1011	$(13)_8$	$(11)_{10}$	0xB
1100	$(14)_8$	$(12)_{10}$	0xC
1101	$(15)_8$	$(13)_{10}$	0xD
1110	$(16)_8$	$(14)_{10}$	0xE
1111	$(17)_8$	$(15)_{10}$	0xF

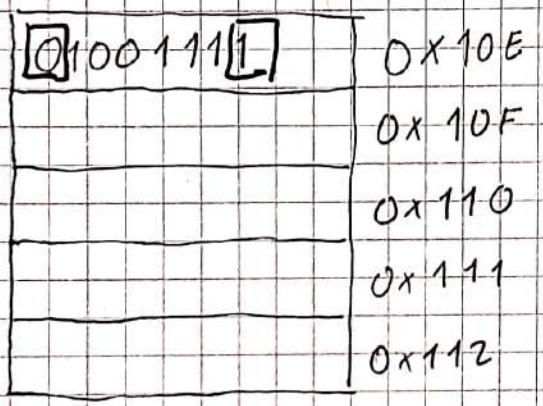
0100 1111 0000 0111

2^{15} ... 2^{0}

Binary

0X4F07 → Hexa decimal

Main Memory



0 = most significant bit
high order end

1 = least significant bit
low order bit

250 →

128	64	32	16	8	4	2	1
1	1	1	1	1	0	1	0

11111010 0xFA

Negative Numbers

- Two's Complement Notation (One's Complement)
- Excess Notation

One's Complement ⟶ 11111010

00000101

Two's Complement ⟶ 11111010

00000101

+ 1

00000110

2^2 2

4+2=6

positive

46+16 1 1 4

64 32 16 8 4 2 1

2^6 2^5 2^4 2^3 2^2 2^1 2^0

0 1 1 1 1 1 1 0 ⟶ 126 , max 127

negative

1 1 1 1 1 1 1 0 ⟶ -2 -2

0 0 0 0 0 0 0 1

+ 1

0 0 0 0 0 0 1 0

2^1 2^0

111

00000000

11111101

+ 1

11111110

00000001

+ 1

-128 ~ 127 (1 byte) ⟶ SIGNED NUMBERS

0 - 255 (1 byte) ⟶ UNSIGNED NUMBERS

Excess Notation

Bit Pattern	Excess 4 Notation
111	= 3
110	= 2
101	= 1
100	= 0
011	= -1
010	= -2
001	= -3
000	= -4
0...7	

$2^k - 1$

Excess-8 Notation

4

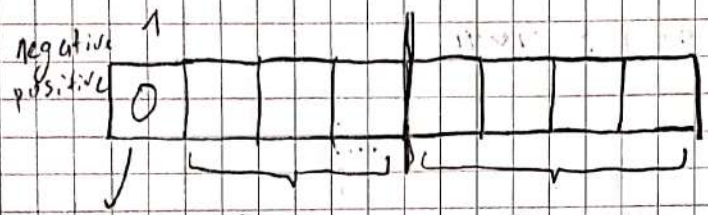
1111
1110
1101
1100
1011

Fractions In Binary

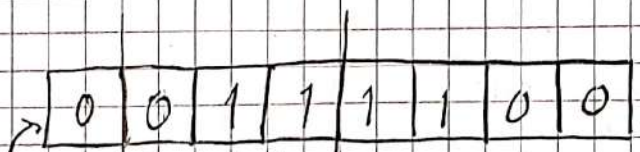
$2^2\ 2^1\ 2^0 \quad 2^{-1}\ 2^{-2}\ 2^{-3}$

$110.101 \rightarrow 6 + \frac{1}{2} + \frac{1}{8} \Rightarrow 6\frac{5}{8}$

Floating-Point Notation



Sign bit — Exponent — Mantissa



Positive

Excess 4 Notation

011 = -1

.1100

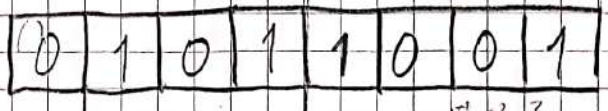
.01100

$2^{-1}2^{-2}2^{-3}2^{-4}2^{-5} = \frac{3}{8}$

Nokta ne kadar kayacak

− sol

+ sağ



\+ 1 . 1.001

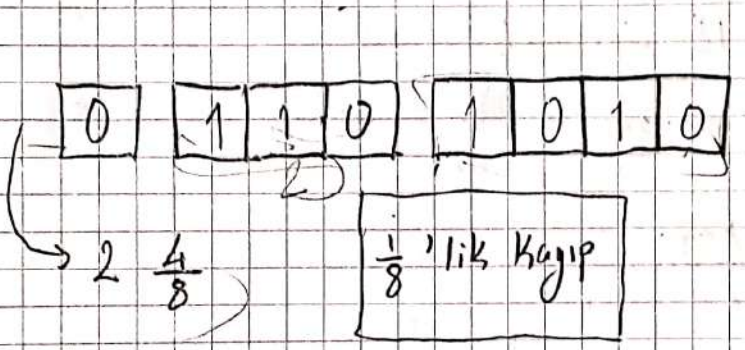
$1 + \frac{1}{8} = 1\frac{1}{8}$

Truncation Error

$2\frac{5}{8}$ Original Representation

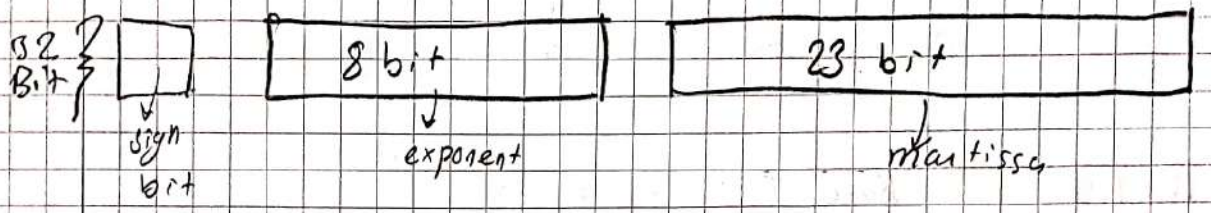
$10.101_{(2)}$ Base Two Representation

1.1010X Raw Bit Pattern



In Normalized Form, Mantissa's first bit is always "1".

Single Precision Floating Point



Double Precision Floating Point

↳ 16 decimal digit

64 bit

Communication Error

Electronical errors.

Parity Bit

(1)

0	1	0	0	1	0	1	1	X
0	1	0	0	1	0	1	1	0

Parity bit

↙ even → Gift

↘ odd → Tek

(2)

1	1	0	0	0	0	0	0	1
1	1	0	0	0	0	0	0	1

Error Correction

An Error - Correction Code

Symbol	Code
A	000000
B	001111
C	010011
D	011100
E	100110
F	101001
G	110101
H	111010

Hamming Distance

000000 A
001111 B
} Hamming Distance = 4

—o—

Decoding the pattern 010100 using the code left side:

Character	Code	Pattern Recieved	Distance
A	000000	010100	2
B	001111	010100	4
C	010011	010100	3
D	011100	010100	1
E	100110	010100	3
F	101001	010100	5
G	110101	010100	2
H	111010	010100	4

Kaelin Color

Representing Information as Bit Patterns

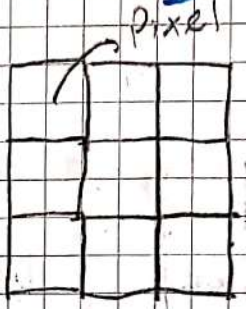
- Character sets → ASCII (1 byte)
- Images
- Sound
- Text
- Numbers
- Video
- Digital Signal
- Instruction Set
- Web Pages
- Programs
- Software

Data Compression

- Lossy Compression
- Lossless Compression

Veri şifreleme
Encode, Decode

Analog vs Digital



Analog daha detaylı.

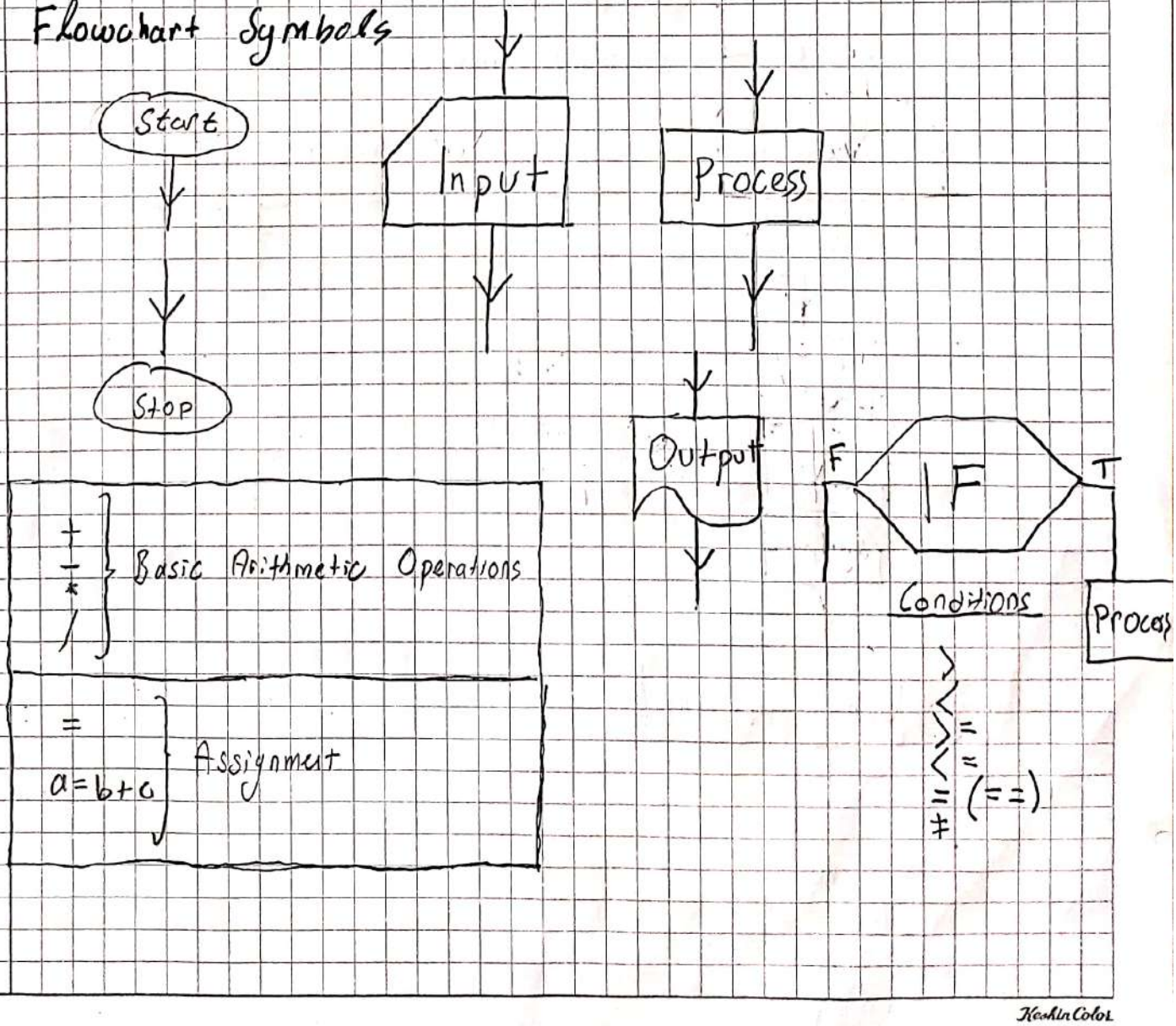
black-white image : 1 bit
gray scale image : 8 bit
RGB color image : 24 bit

ALGORITHMS

An algorithm is an ordered set of unambiguous executable steps that define a terminating step.

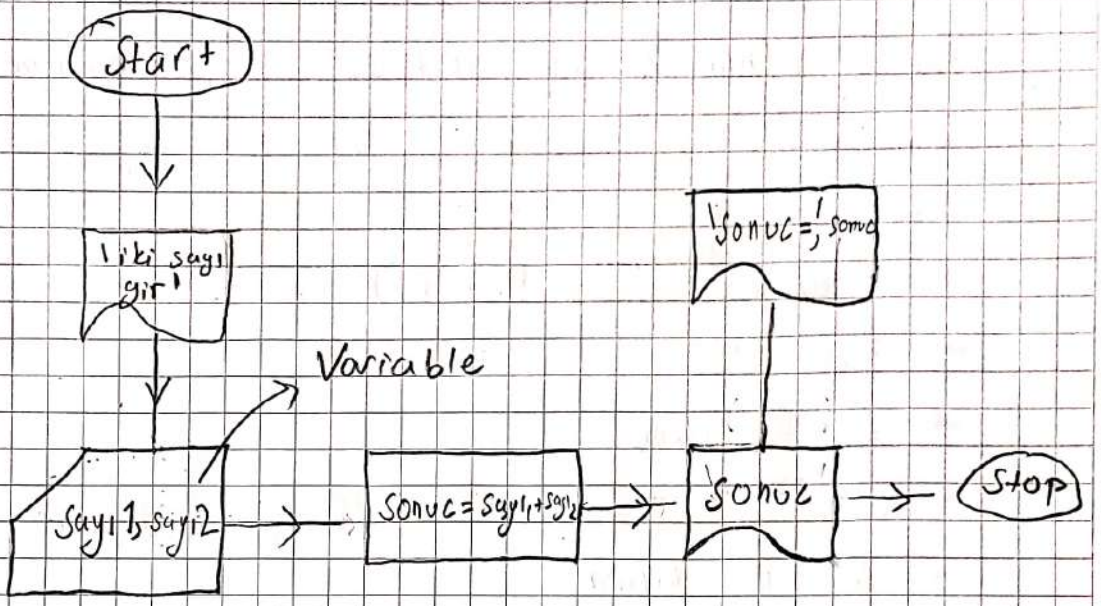
- Flowchart: (Algis Diyagramı)
- Pseudocode
- List of items
- Code
- Oral Presentation

Flowchart Symbols



Question

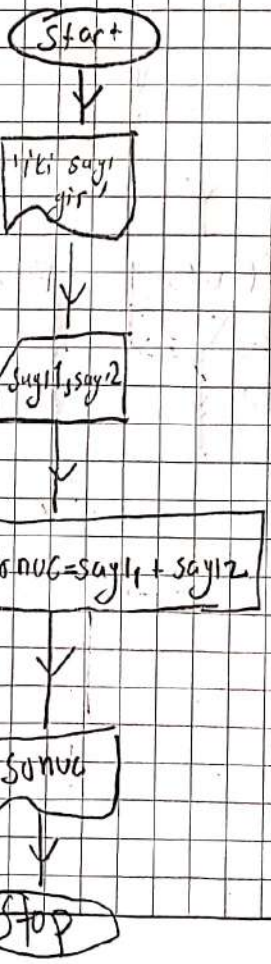
Flowchart of $Number_1 + Number_2$



a = 3 ⟶ constant number

Variable

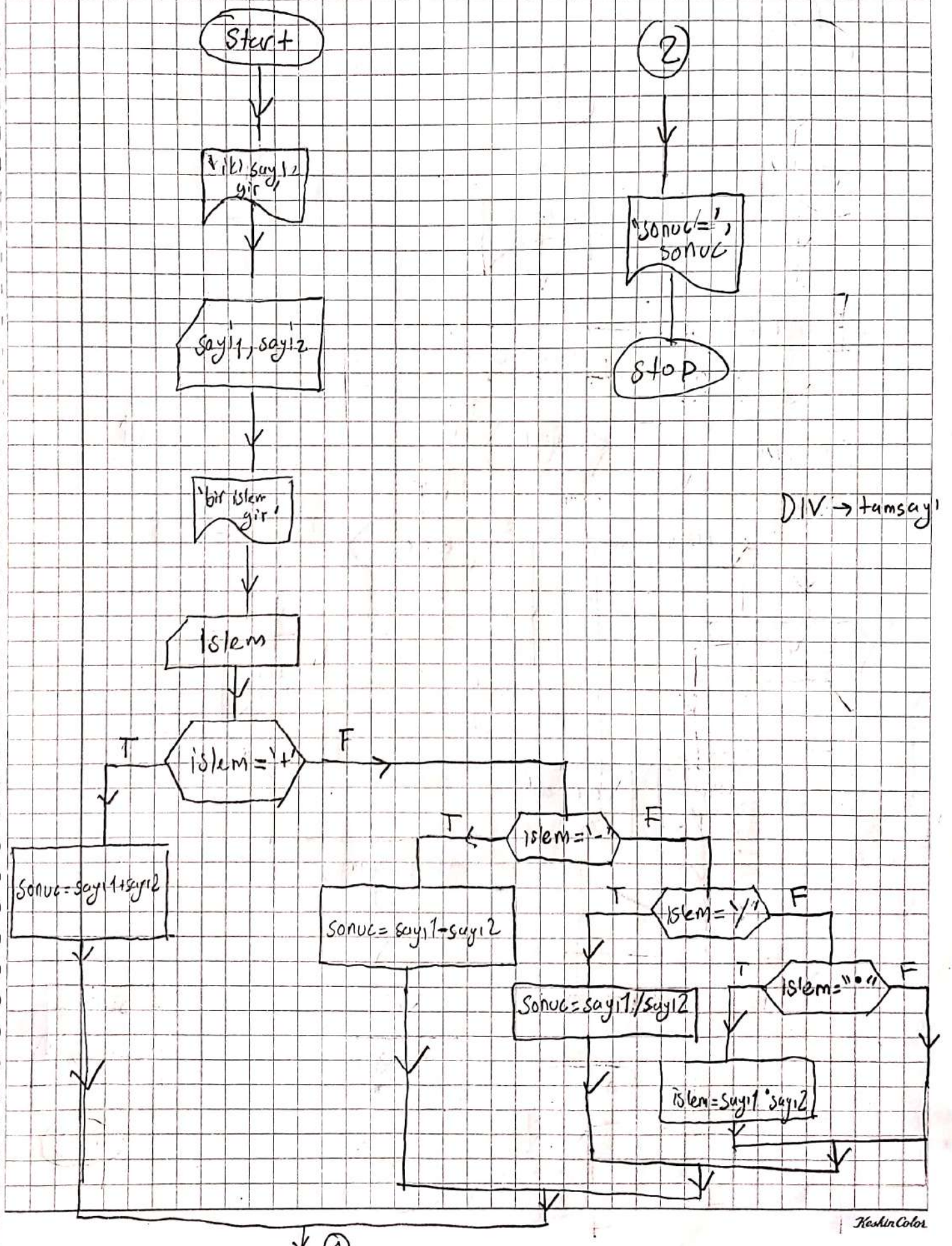
'a' character



Keskin Color

Question

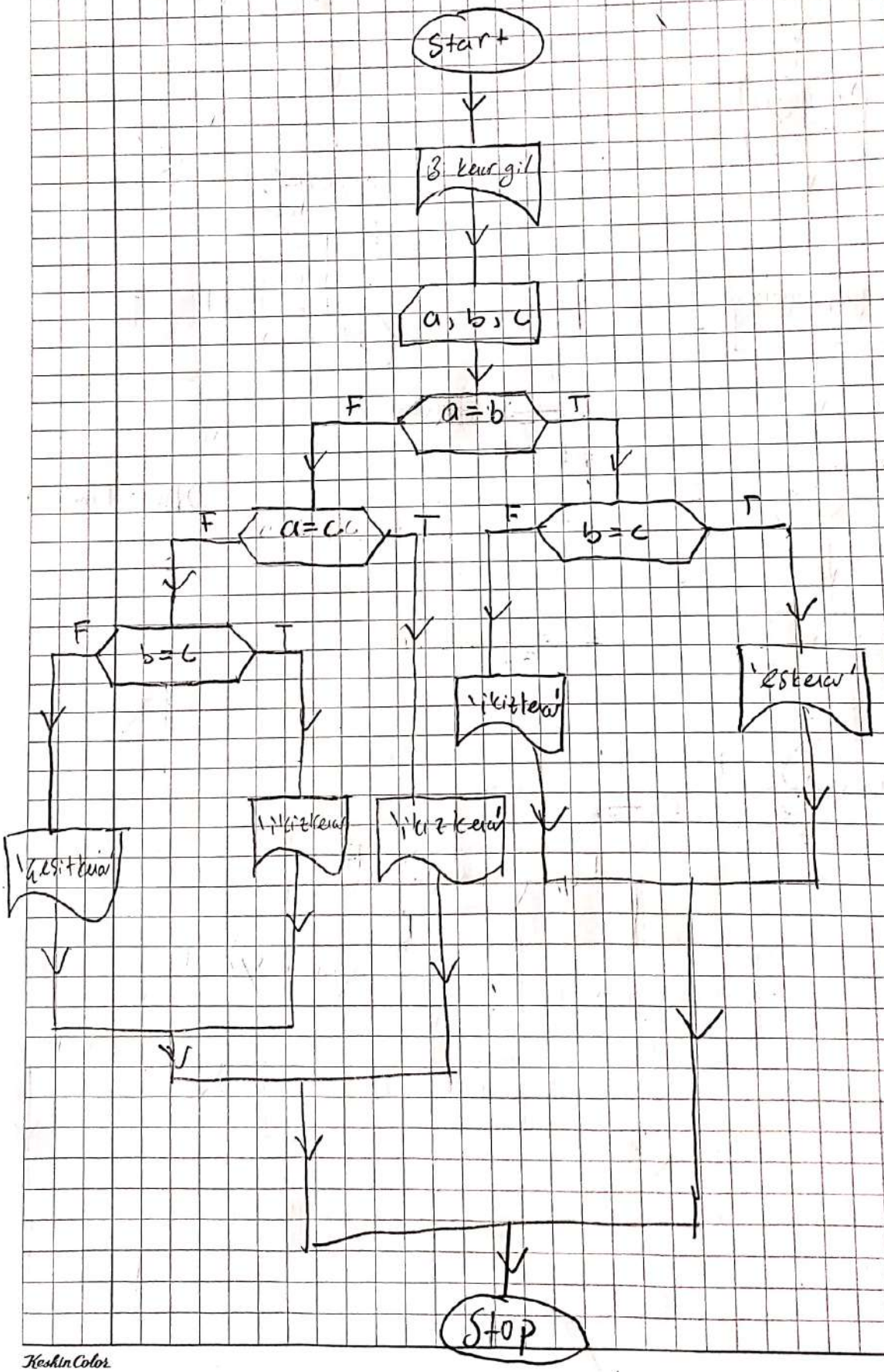
flowchart of basic arithmetics



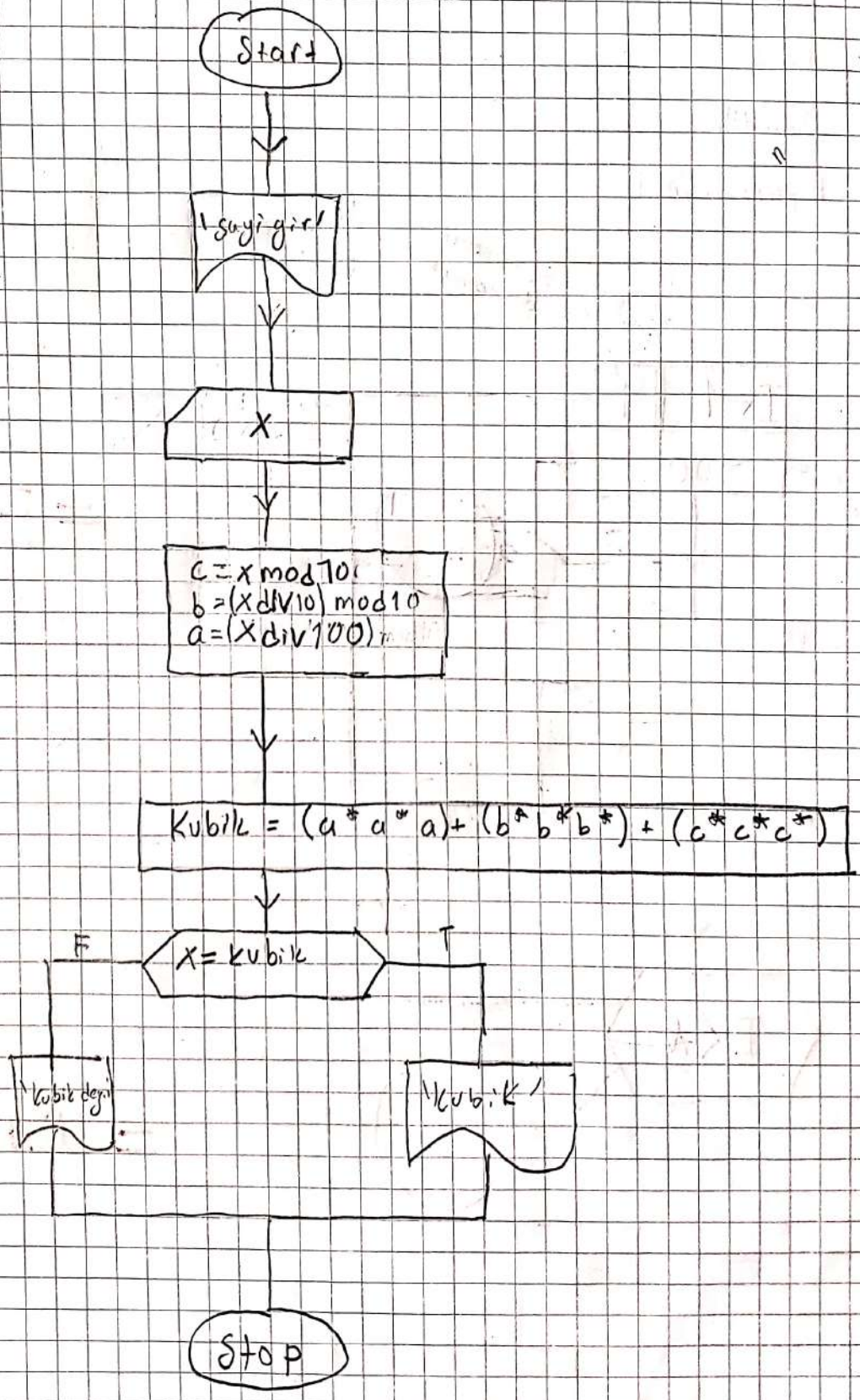
DIV → tamsayı

KeskinColor

Q: 3 kenarı verilen üçgenin tipini bulan algoritma + flow chart

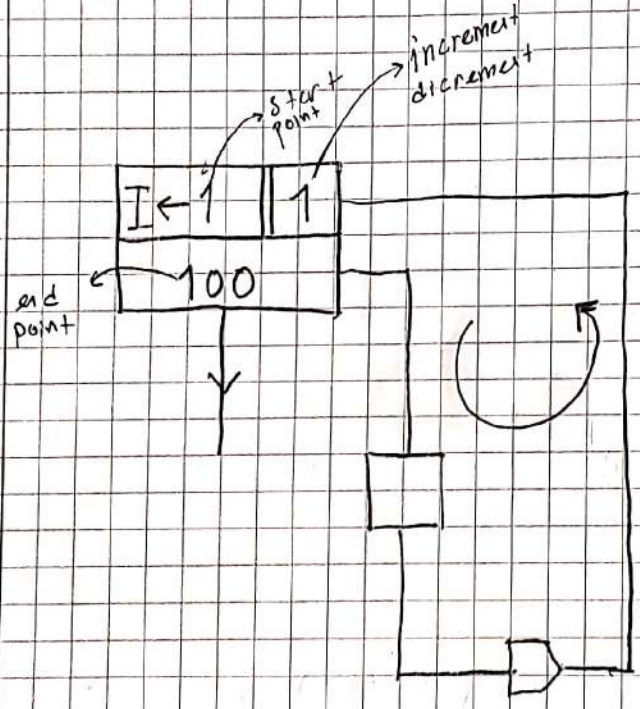


8/ Kübik sayı =? $(abc = a^3 + b^3 + c^3)$

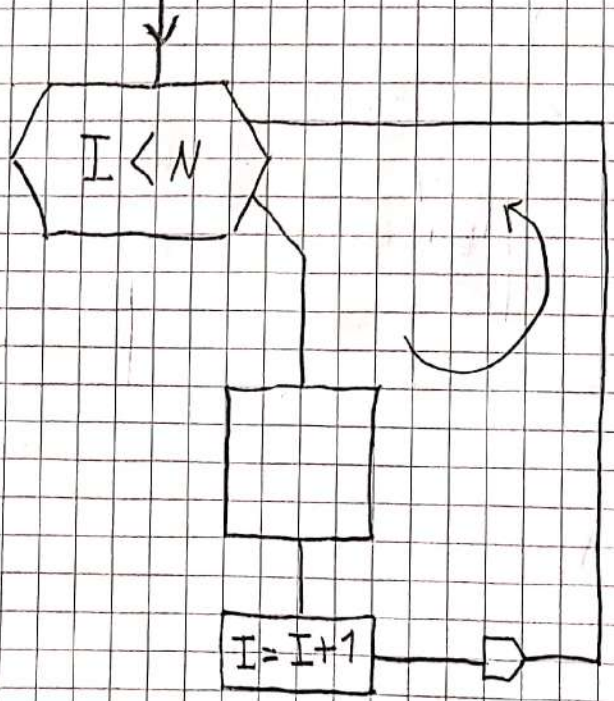


LOOPS

- For
- While
- Do... While
- Repeat... Until

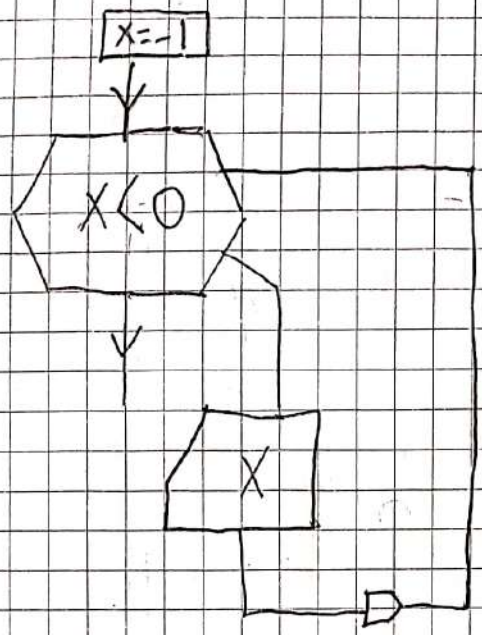


= FOR



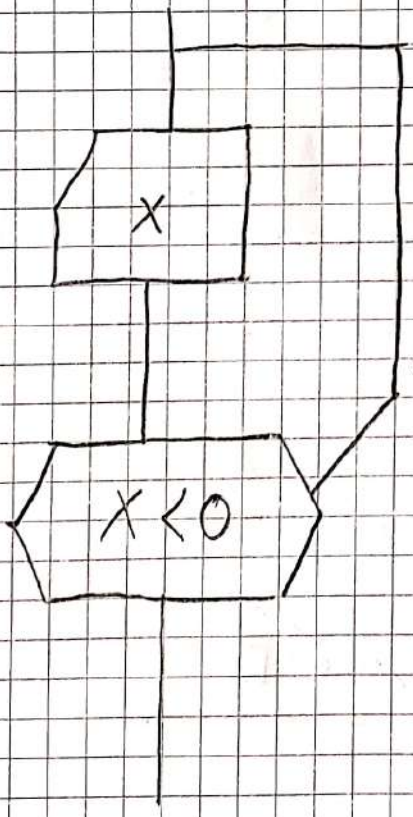
= WHILE

a



→ kontrol etme amaçlı

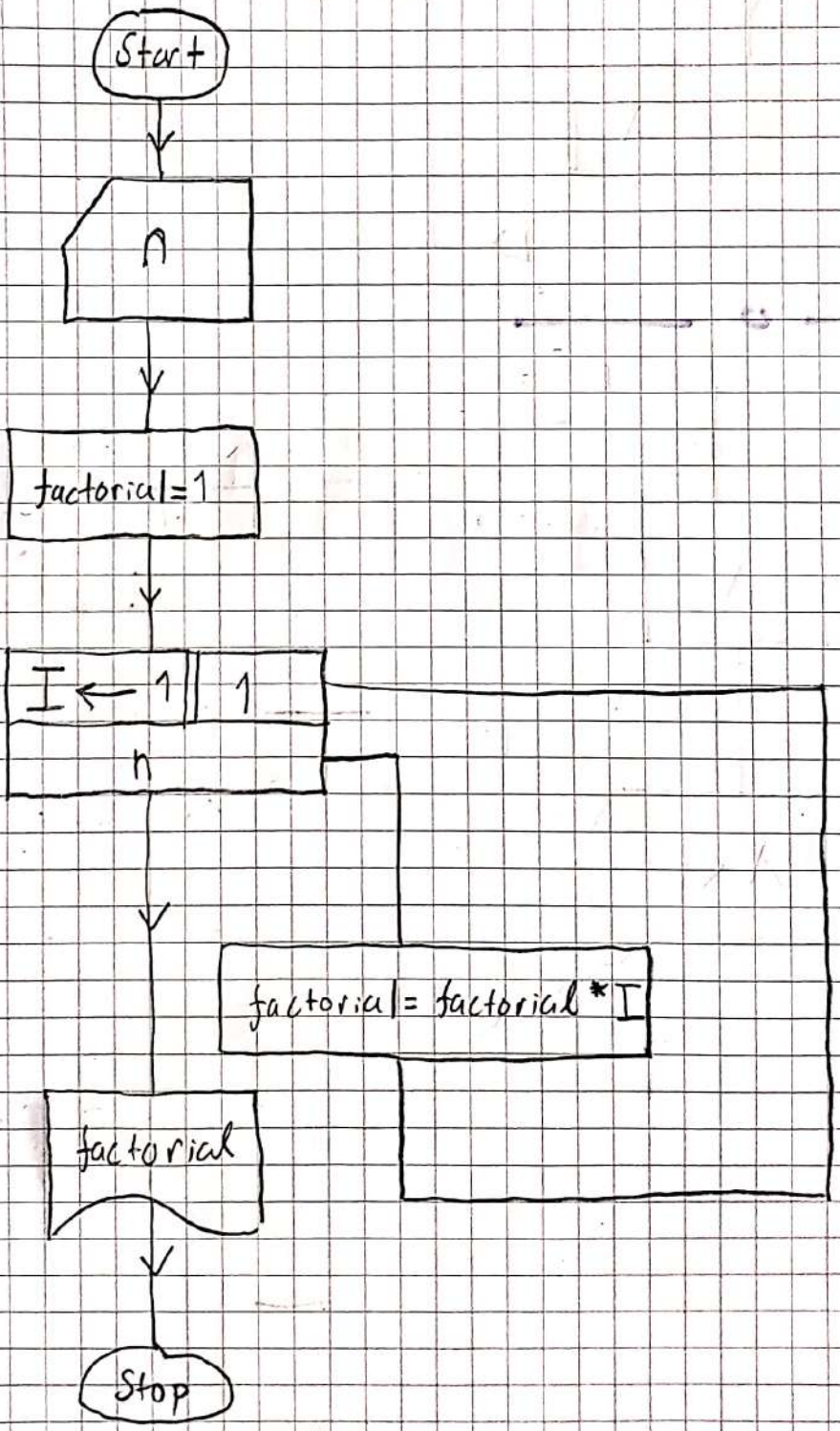
b



= DO... WHILE

Q/ flowchart of calculator of n!

Initial Value = 1



n=0 iken döngüye girmez faktöriyel = 1 olur ($0! = 1$)

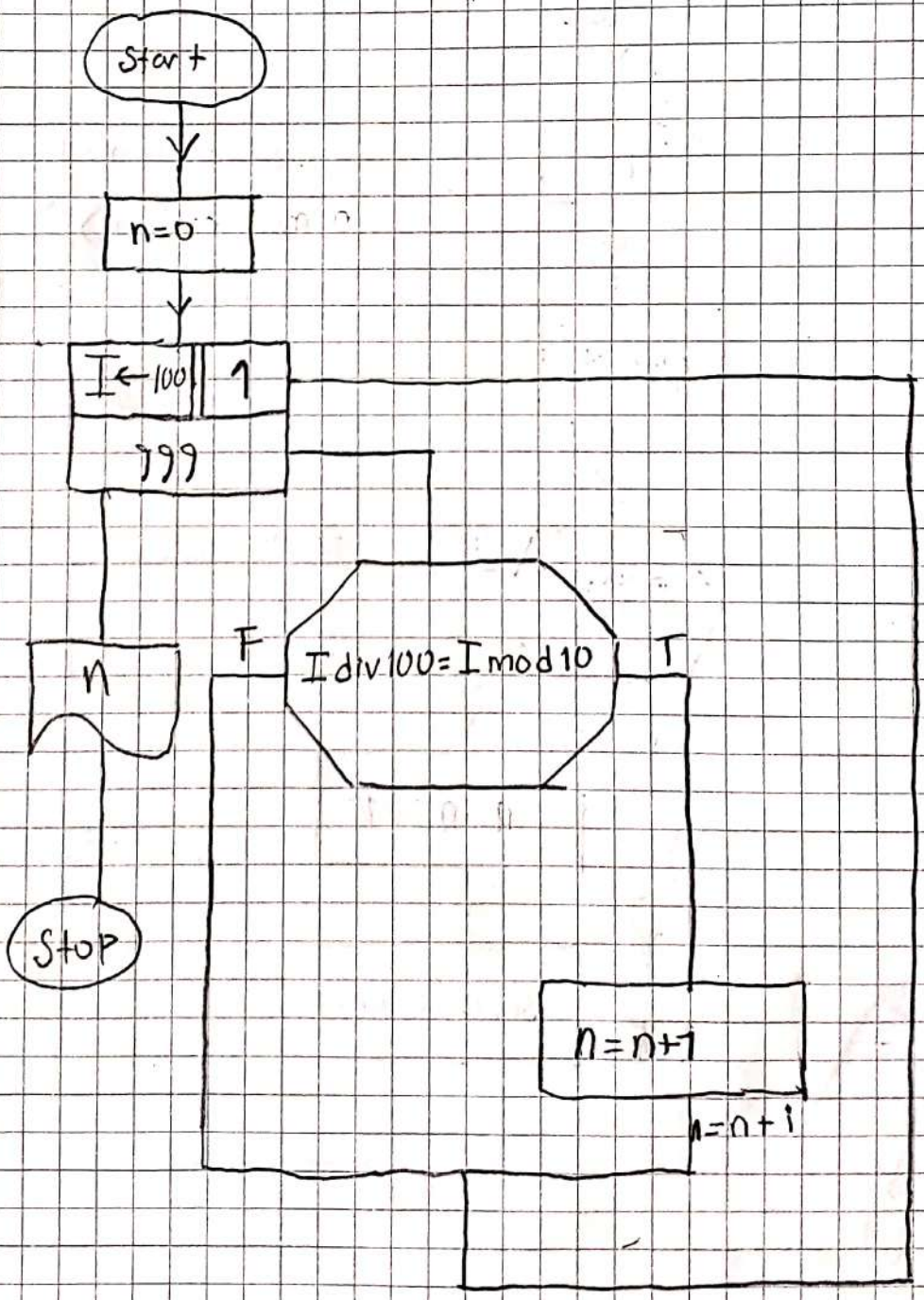
n=1 iken ve I ← 2 iken döngüye girmez ve faktöriyel = 1 olur

Palindrome Numbers

100
101 *
102
⋮
998
999 *

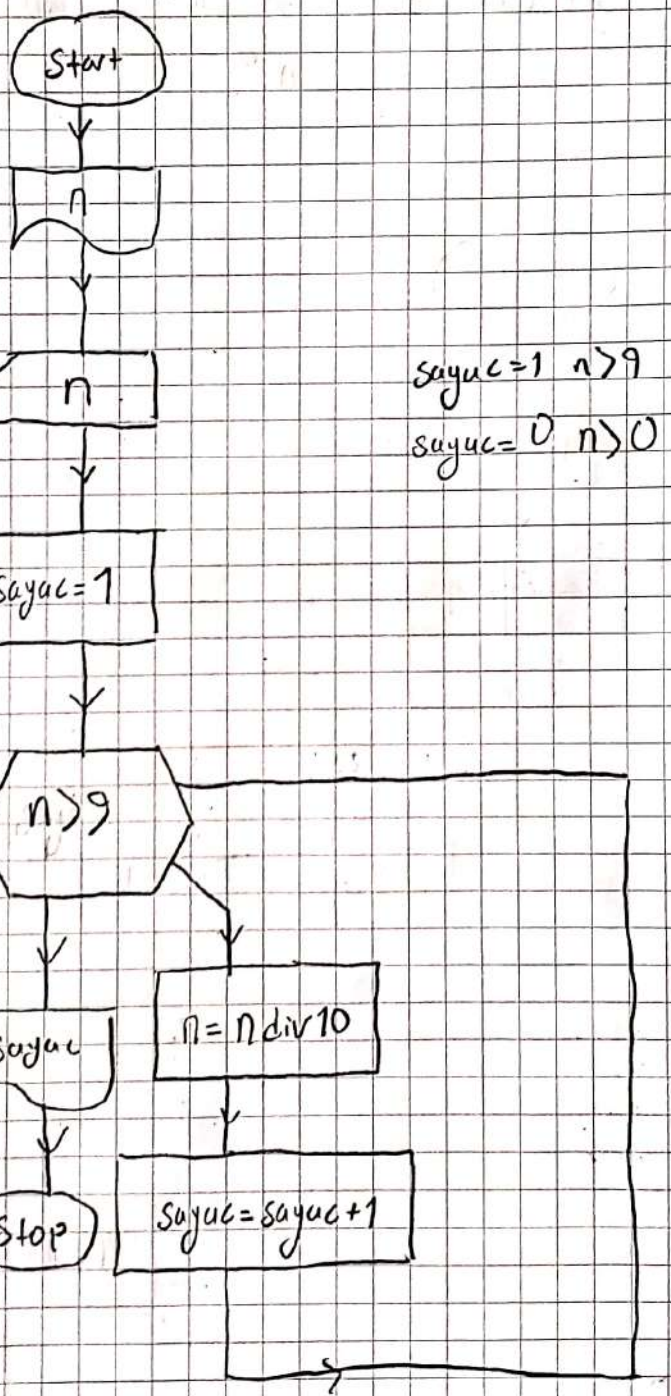
$n \in [100, 999]$

Q
"flowchart of all palindrom numbers n"



Q/

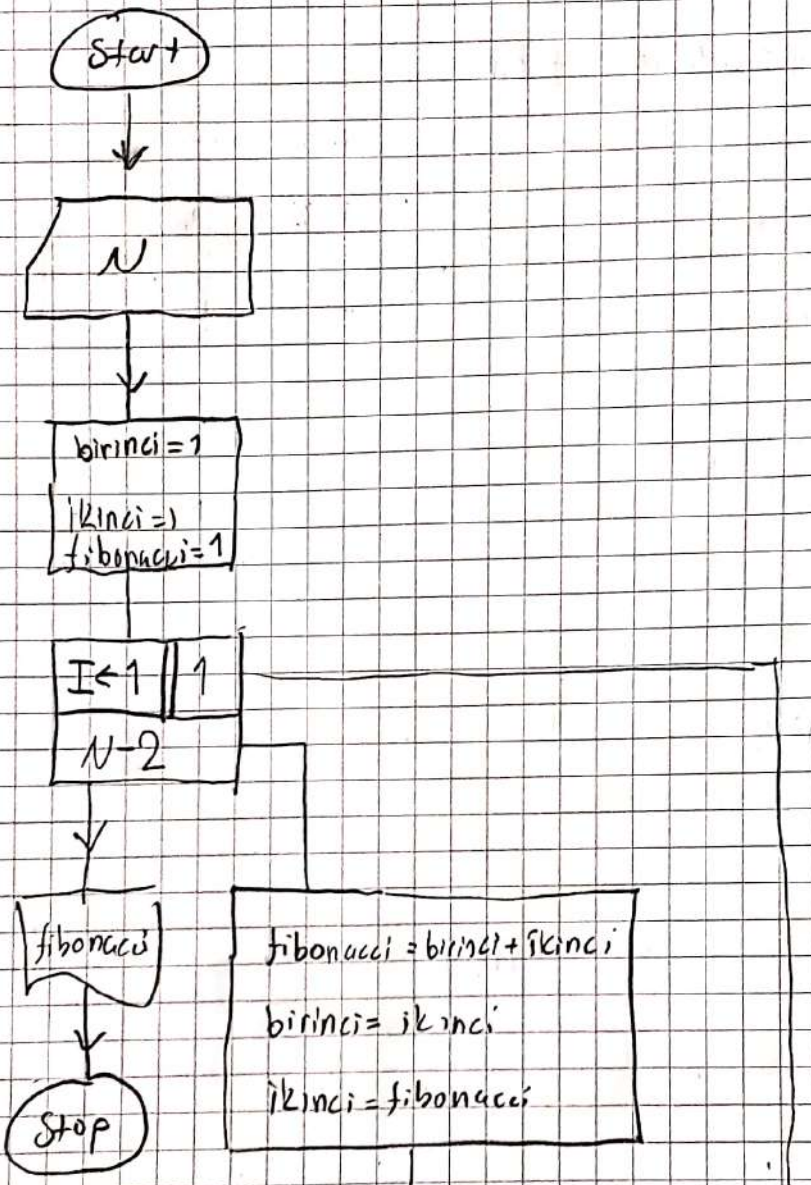
Verilen sayının kaç basamaklı olduğunu bulan flowchart



sayac=1 n>9 için 0'ı kapsar

sayac=0 n>0 için 0'ı kapsamaz

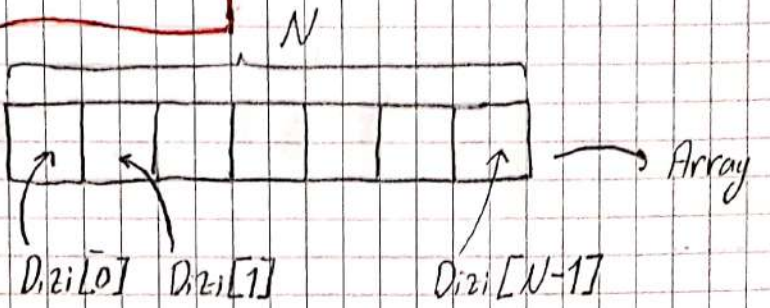
negatifler için de göz



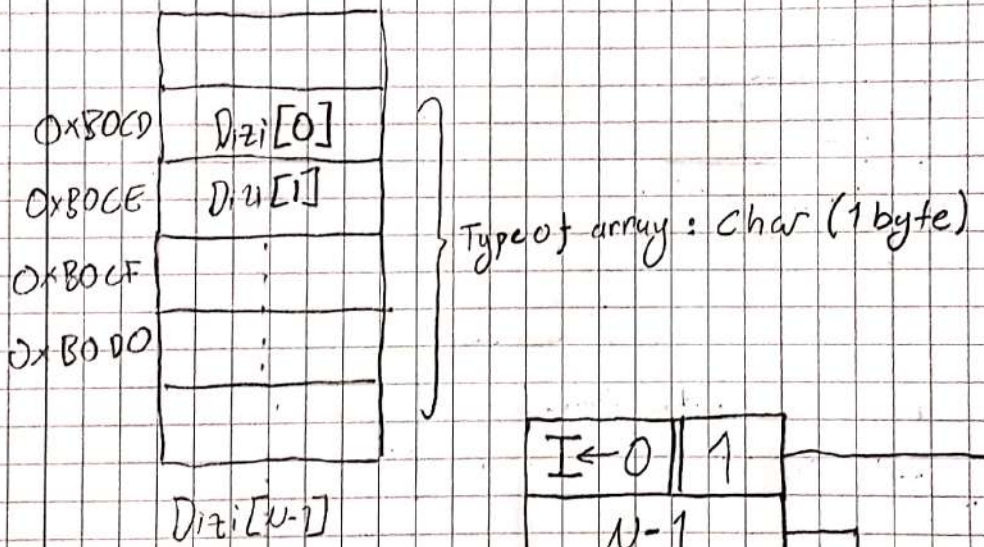
I	b	i	f
1	1	1	1
2	1	2	3
3	2	3	

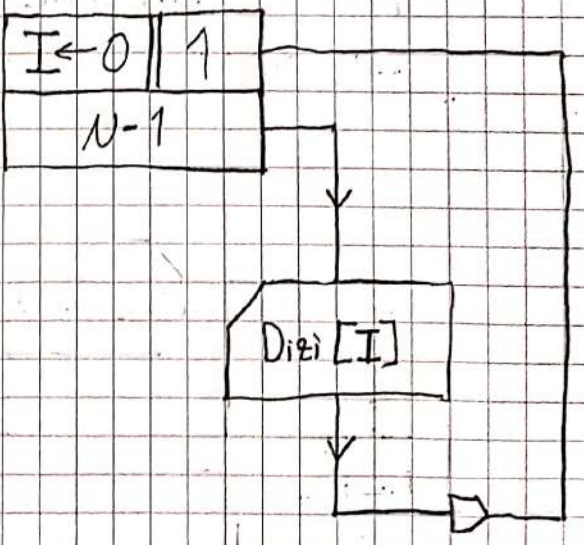
character 1byte
int 4byte

Arrays



Memory Adress



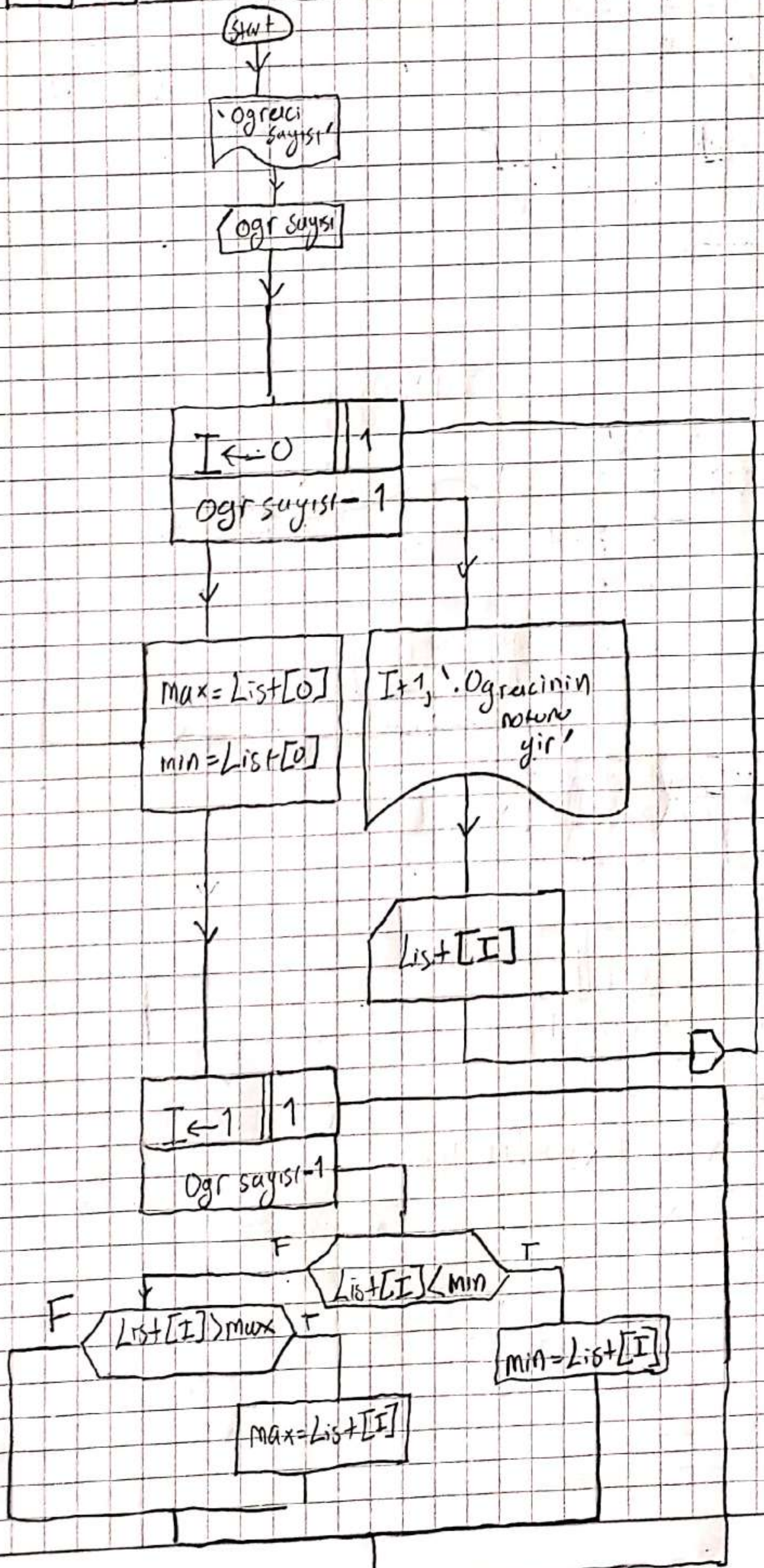


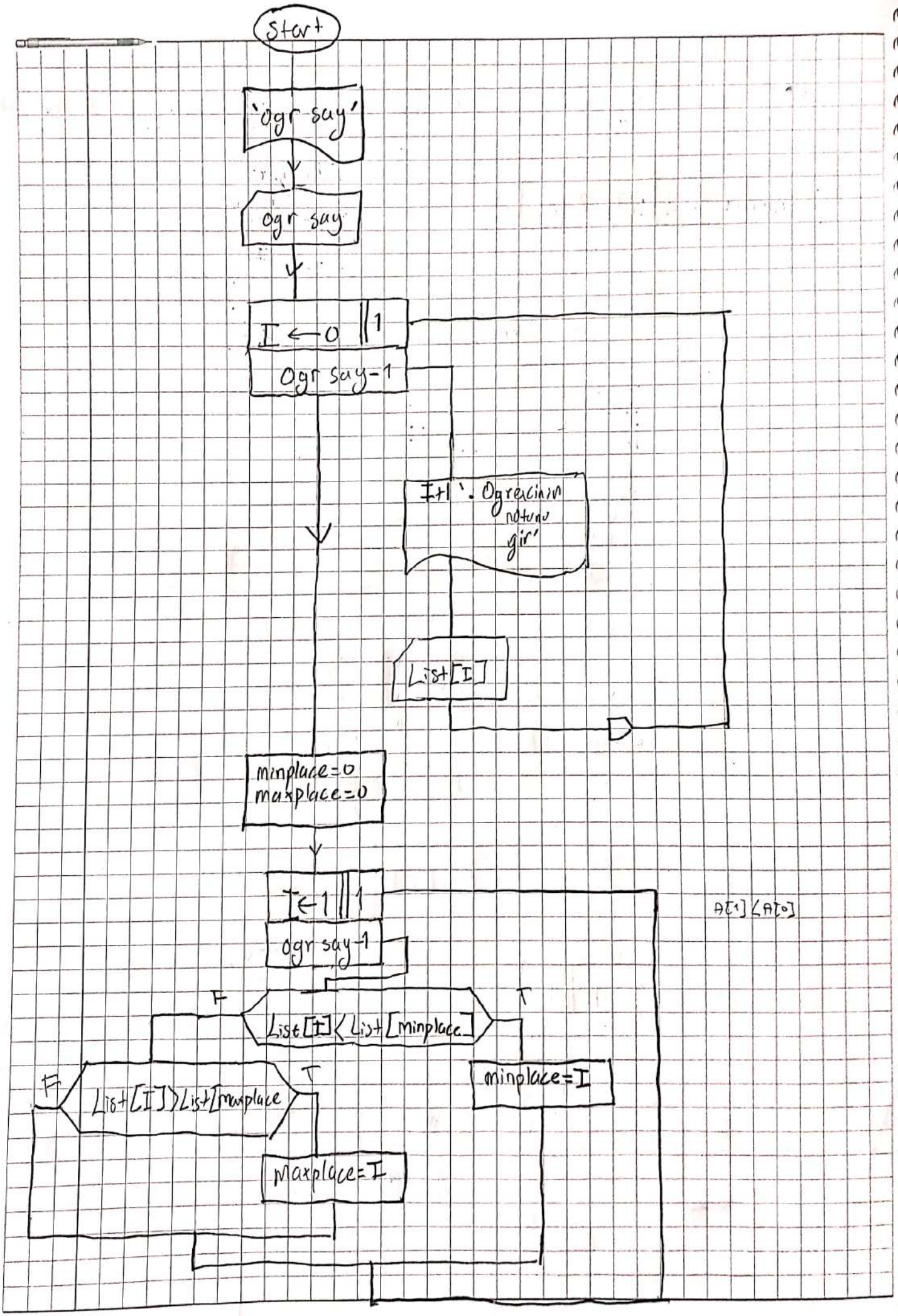
Array Operations

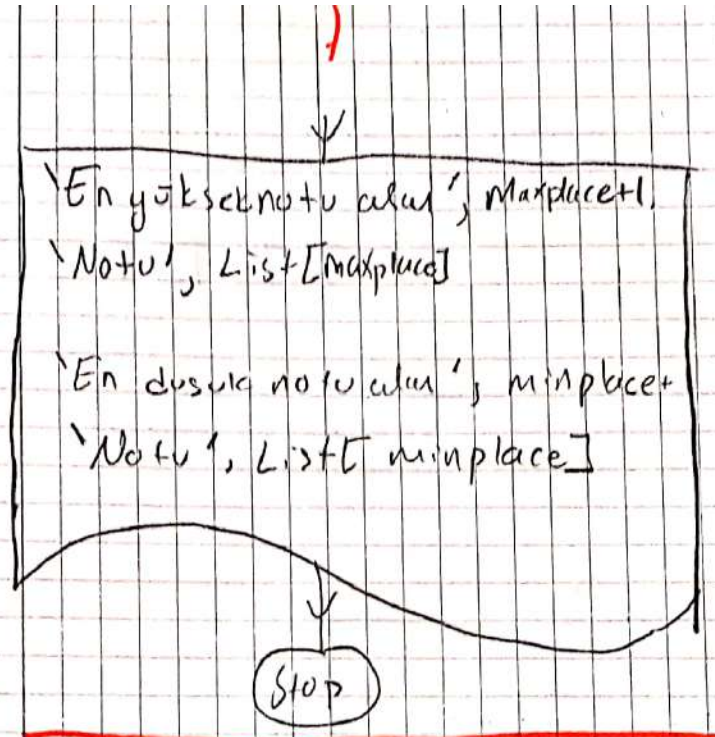
- Sorting
- Storing
- Insert
- Delete
- Search
- Update
- Minimum
- Maximum

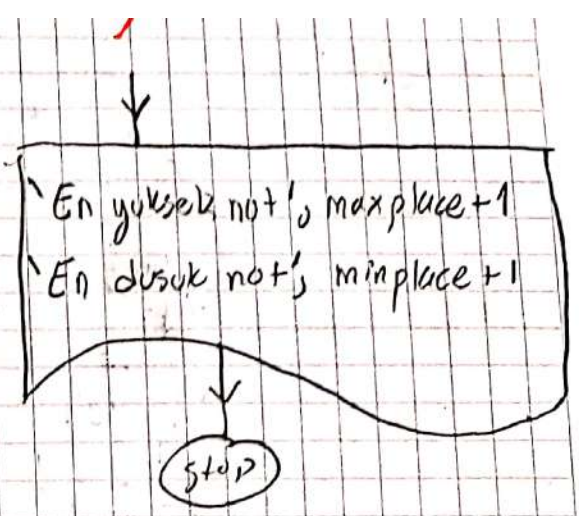
Q/ Given the mark list, find that students who have min and max marks.

0	1	2	3	4
72	38	41	82	33



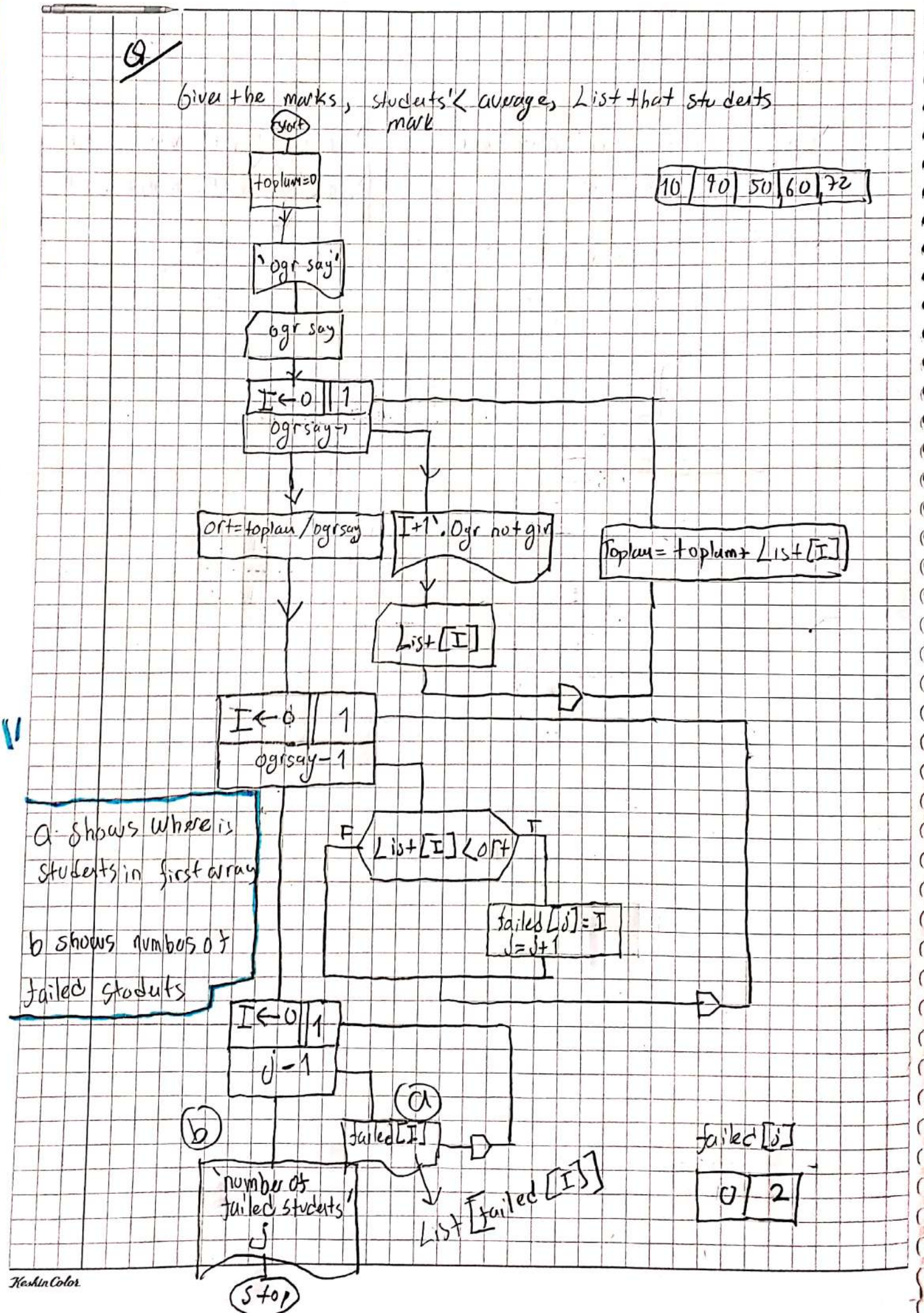



'En yüksek notu alan', maxplace+1
'Notu', List[maxplace]
'En düşük notu alan', minplace+1
'Notu', List[minplace]
Stop


'En yüksek not', maxplace+1
'En düşük not', minplace+1
Stop

Q/

Given the marks, students' average, List that students mark



Like, Object Code

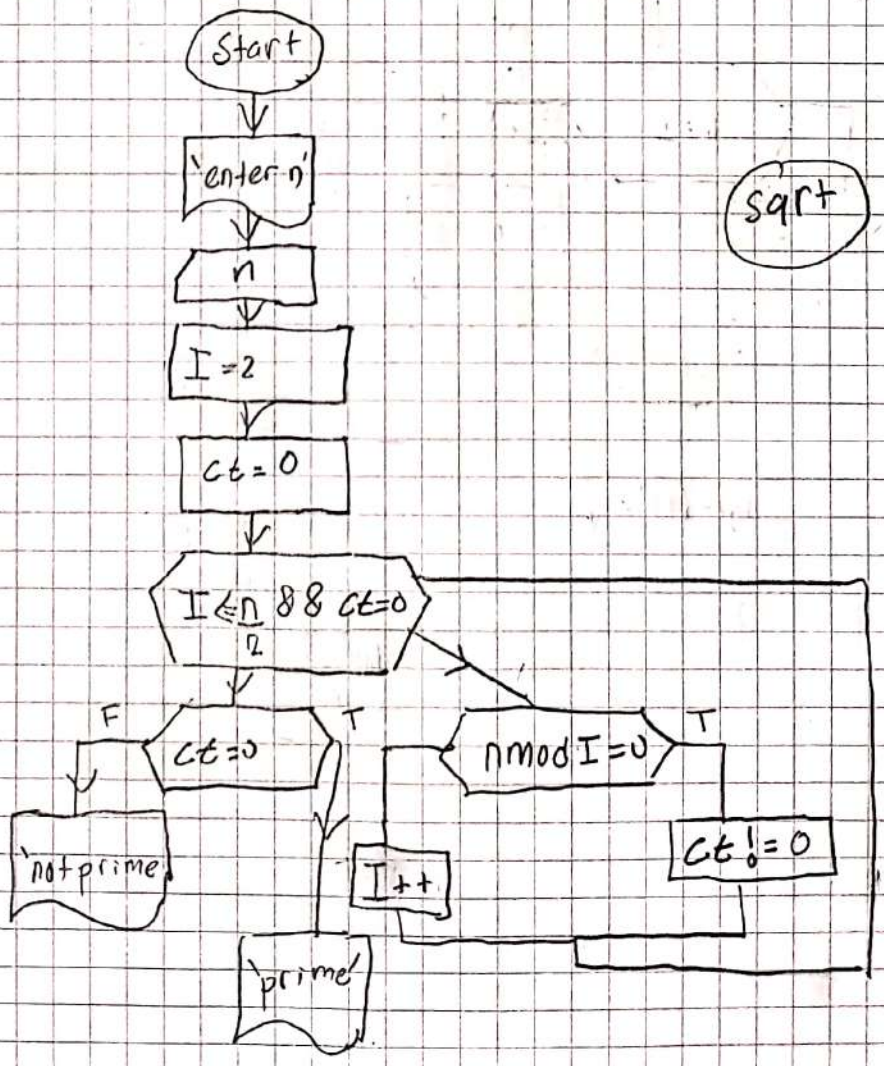
Integrated Development Environment

Programming Language : C

↳ Editor
Compiler
Debugger
Execute / Run

```
#include<stdio.h>
int main() {
    printf("Welcome to C programming");
    return 0;
}
```

Flowchart of the given number is whether prime or not.

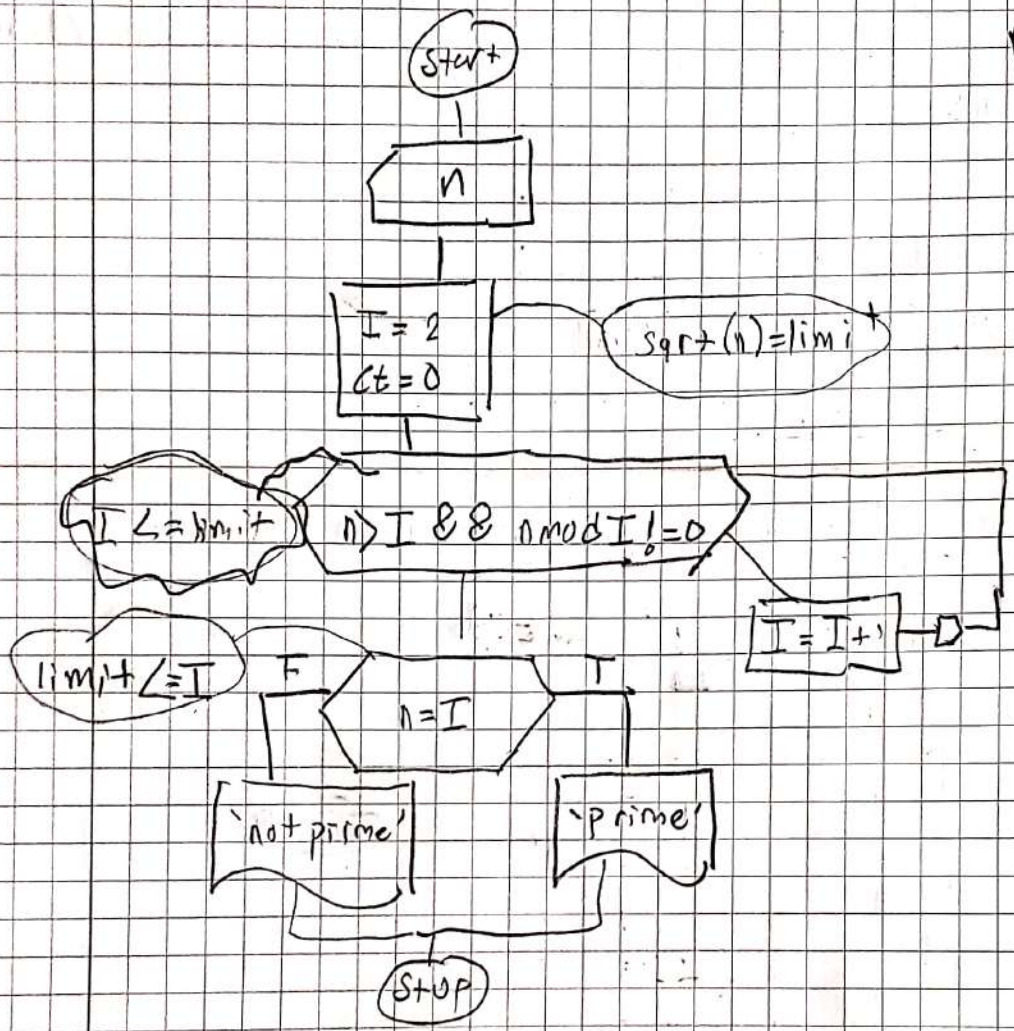


sqrt

n/1
n/2
n/3
n/4

2 flowchart of the gives number is whether prime or not

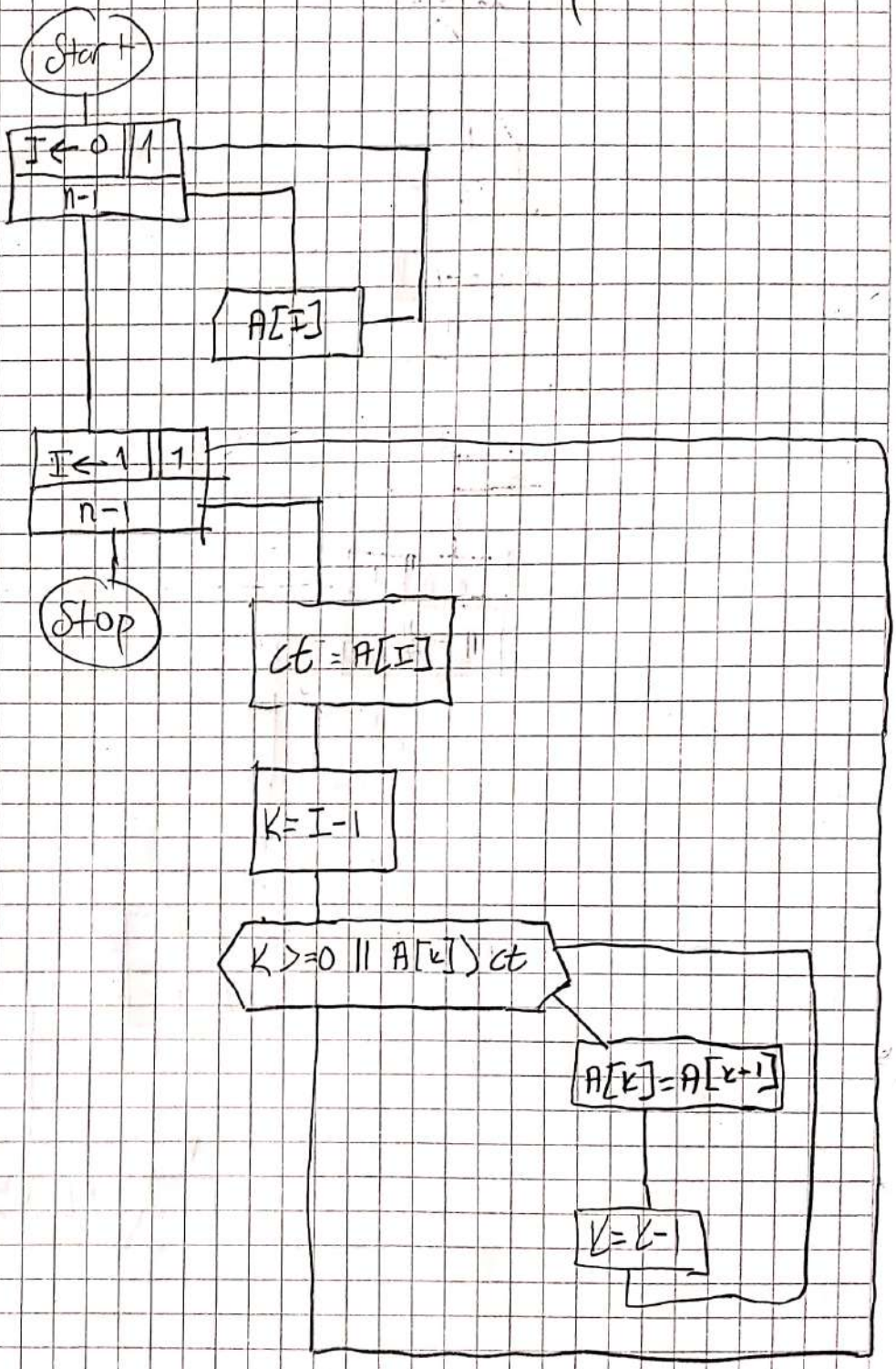
PSRL



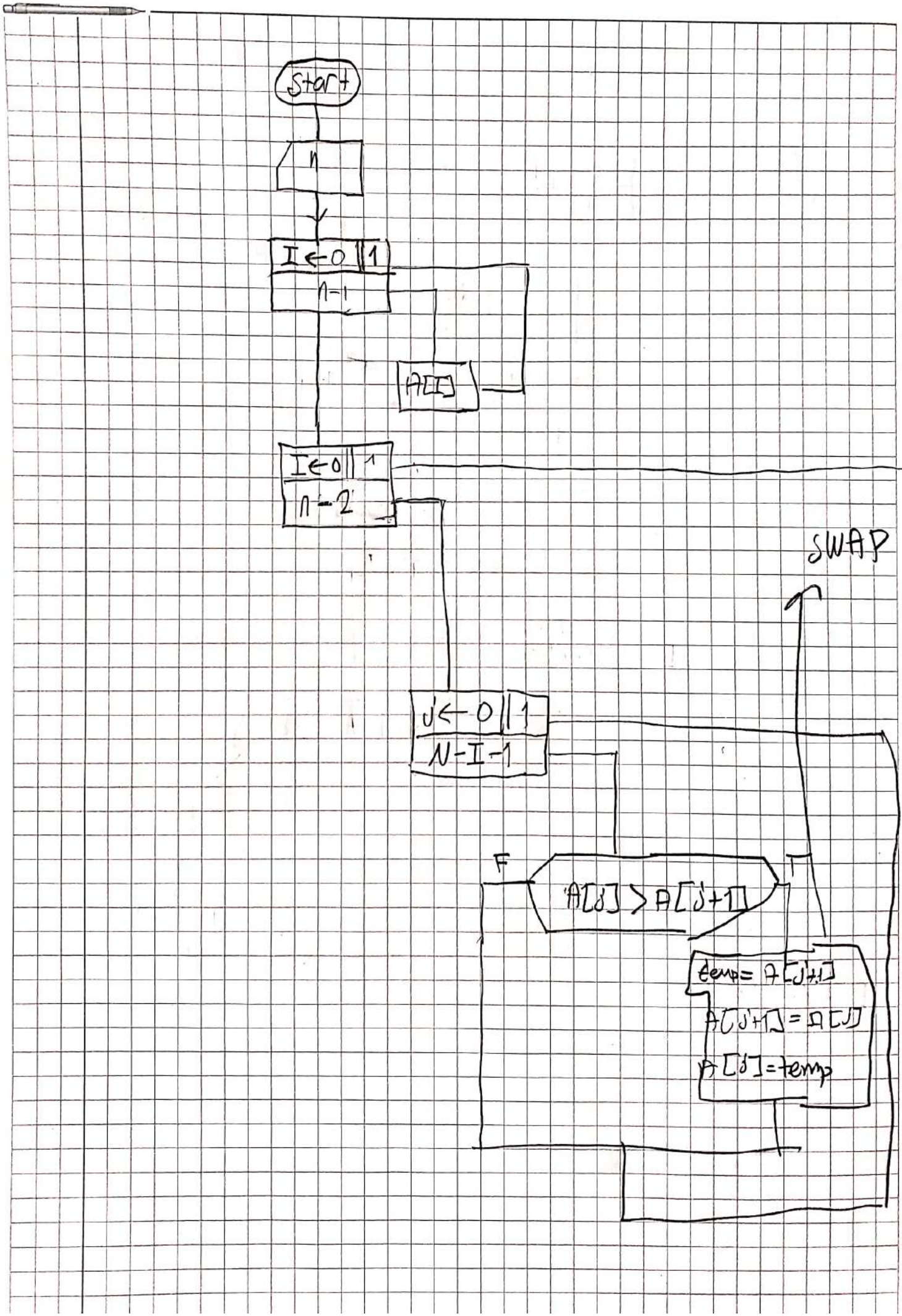
17 sqrt(17) = 4+

2
3
4

Sorting



Bubble Sorting


Start
n
I ← 0 || 1
n-1
A[I]
I ← 0 || 1
n-2
SWAP
j ← 0 || 1
N-I-1
F
A[j] > A[j+1]
temp = A[j+1]
A[j+1] = A[j]
A[j] = temp

Variable Types In C

unsigned { char → 1 byte
int → 4 byte
float →
double →

Short int → 2 byte, long int → 8 byte

Control Flow

- Conditional Branching

```
if(x)
    statement 1;
statement 2;
```

```
if(x)
    statement 1;
else
    statement 2;
statement 3;
```

```
if(x) {
    statement 1;
    statement 2;
}
else {
    statement 3;
    statement 4;
}
statement 5;
```

Compairsion Expressions

<
>
<=
>=
==
!=

Array Declaration / Initialization

```
int array[100];
```

↓

array[0] ..... array[99]

Looping

- while
- for
- do... while

```
while (statement)
statement1;
```

```
while (statement) {
    statement1;
    statement2;
}
```

```
int x=3, y=7;
while ((x<y) && (y<z)) {
x++;
z--;
}
```

```
for (expression1; expression2; expression3;)
statement1;
```

```
for (i=0, x=1; i<=N; i++) {
    x = x+i;
}
```

```
for (exp1; exp2; exp3;) {
    statement1;
    statement2;
}
```

```
int i;
int x=1;
for (i=1; i<=N; i++)
    x = x*i;
```

ASCII UNICODE EBCDIC

```
do
    statement;
while(expression);
```

```
printf("Enter a sentence:\n");
do{
    ch = getchar()
    if(ch==' '){
        num_of_spaces++;
    }
}while(ch!='\n');
```

(boşluk silme prog yaz)

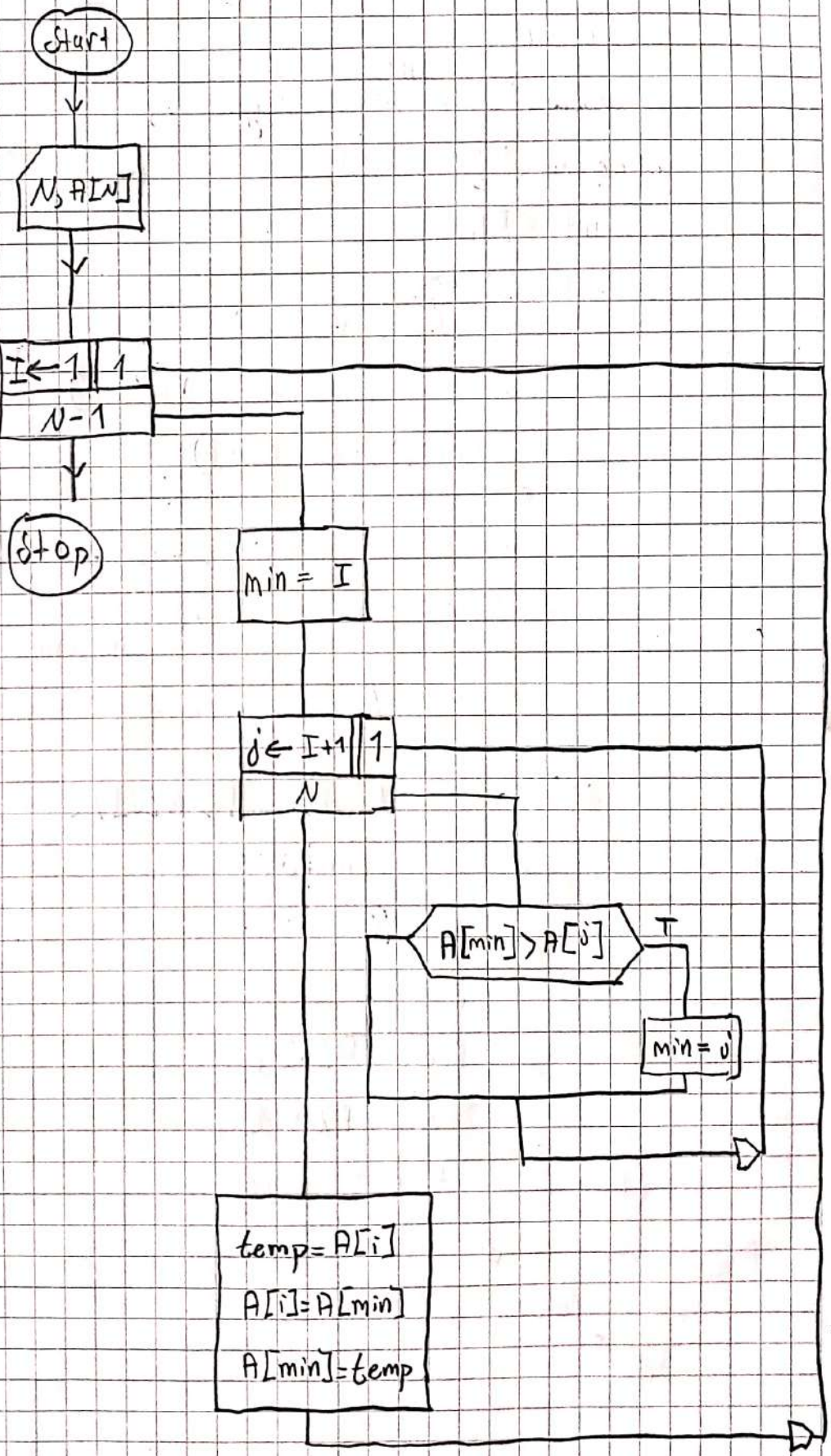
'0' → 48

'1' → 49

1	2	3
49	50	51

'1' - '0' → 1

Selection Sort



Selection sort in C

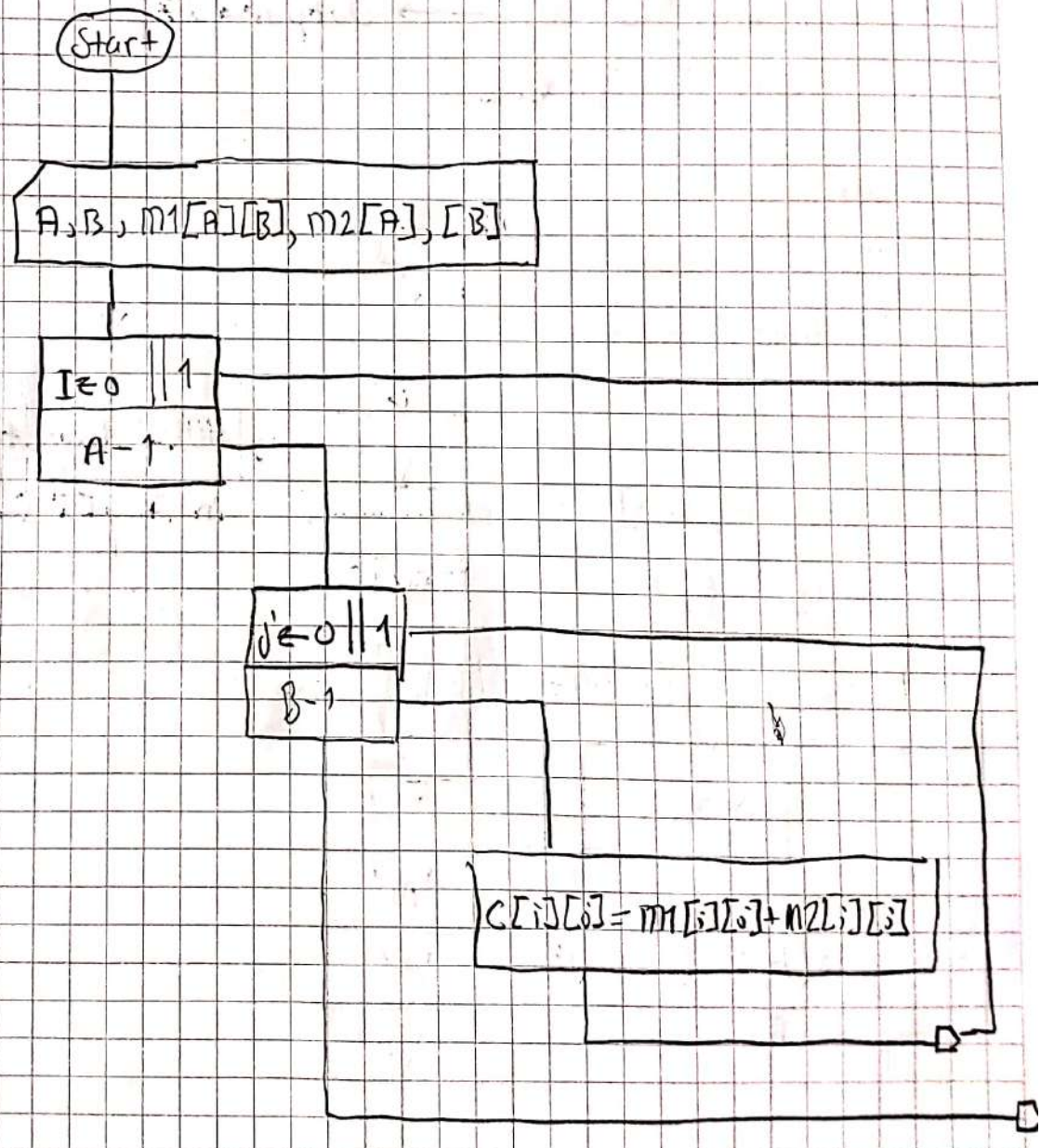
```
#include<stdio.h>
int main() {
        int N;
    printf(" Enter Array's Leght: ");
    scanf("%d", &N);
    int A[100];
    int i,j;
      for(i=1; i<=N; i++){
        printf(" enter A[%d]:", i);
        scanf("%d", A[i]);
    }
        int temp;
     for(i=1; i<=N-1; i++) {
                        int min = i;
            for(j=i+1; j<=N; j++) {

              if( A[min] > A[j]) {
                min=j;
              }
          }
          temp = A[i];
          A[i]= A[min];
           A[min]= temp;
        }
    return 0;
}
```

A[5] ??

Multidimensional Arrays in C

int m1[3][2];

0,0	0,1
1,0	1,1
2,0	2,1




```
for (i=0, j=0; i<A-1, j<B-1; i++, j++) {
    c[i][j] = m1[i][j] + m2[i][j];
}
```

→ Array of Characters

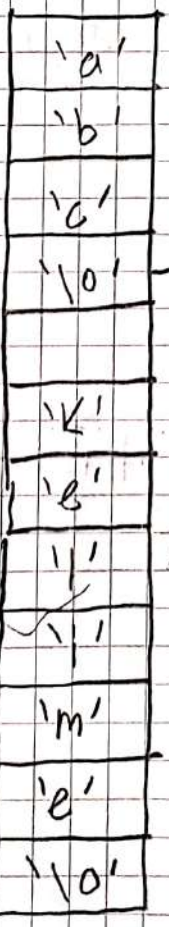
Characters and Strings

```
Char a;
a='a';
printf("%c", a); ⟶ Output = a
printf("%d", a); ⟶ Output = 97
scanf("%c", &a);
```

String

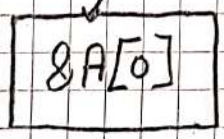
"abc" ⟶

"Kelime" ⟶



→ NULL character in ASCII Table → 0

```
Char A[100]
scanf("%s", A);
```




```
gets(A); (i love it)
puts(A);
```

→ ekrana yazdirma

```
printf("%s", A);
```

(NULL disindaki bosluk dahil her sey char)

NULL = Each string should end with NULL character.

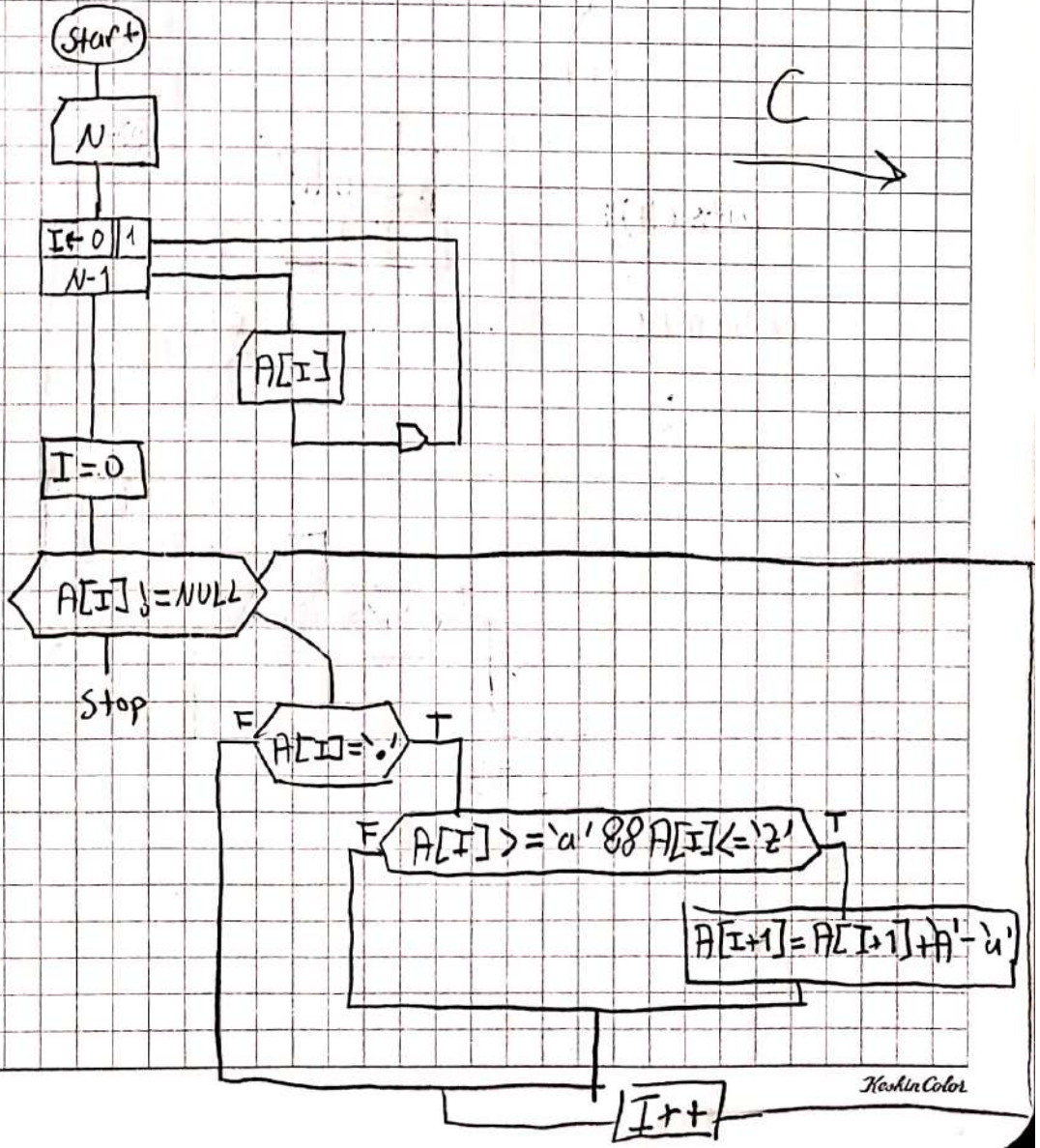
```
char b,c,d;
b=1;
b='1';
c='a';
c=b;
```

```
char word[]="word";
char Kelime[50]="example";
Kelime="wrong";
```

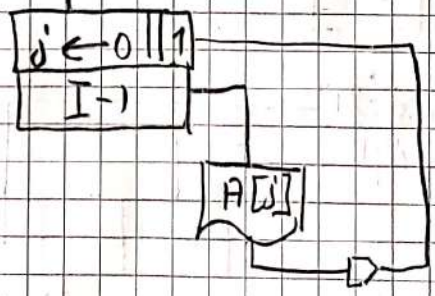
```
int i=0;
While(A[i]!=NULL){
A2[i]=A[i];
i++;
}
```

Q/ Verilen paragraftaki cümlenin ilk harfi küçük ise büyük harfe dönüştüren algoritmanın flowchart'ı ve C kodu.

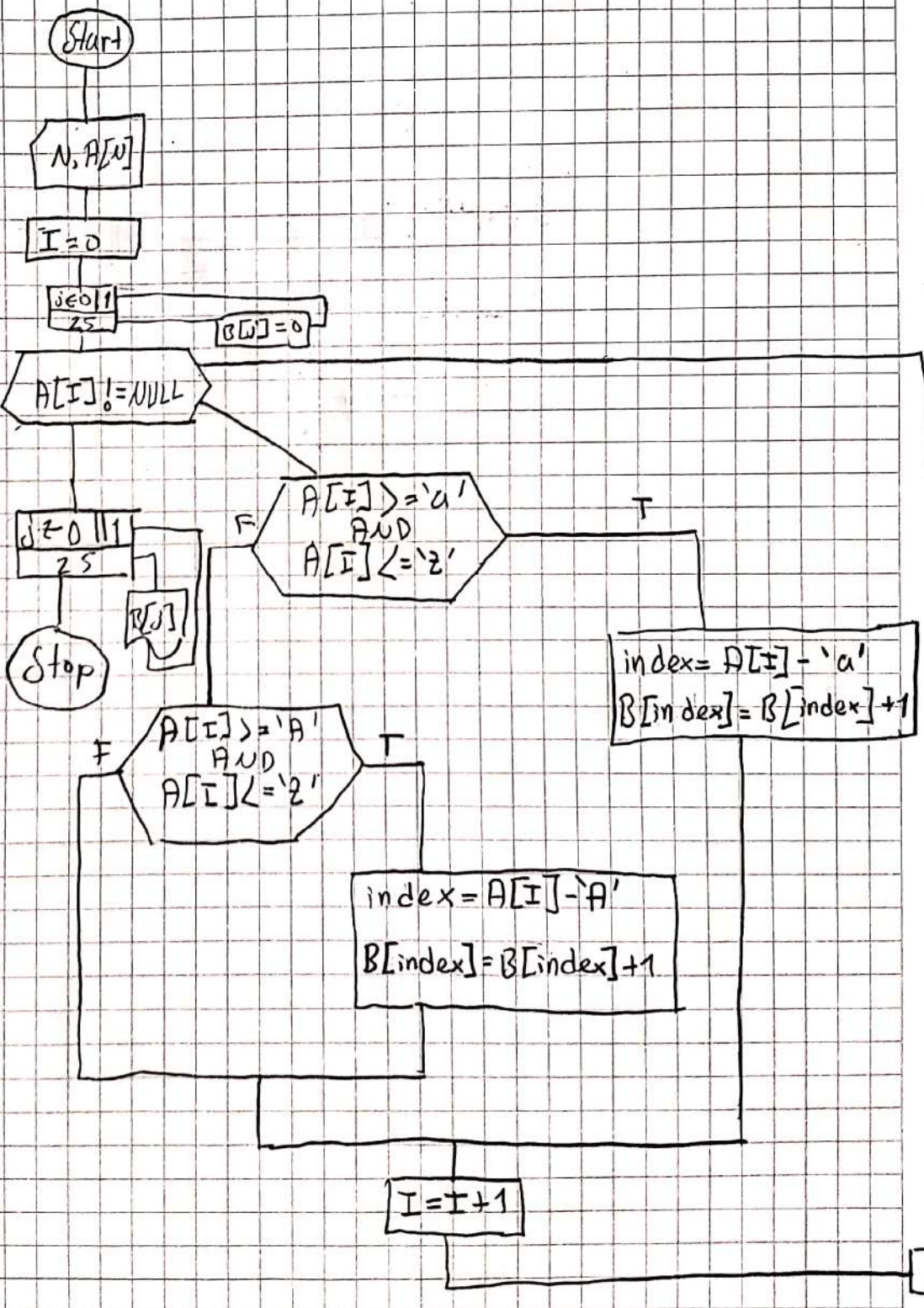
C →



```
#include<stdio.h>
#include<string.h>
int main() {
    char A[200];
    scanf("%s", A);
    int i=0;
    while (A[i] != NULL) {
        if(A[i]='.') {
            if (A[i]>='a' && A[i+1]<='z') {
                A[i+1] = A[i+1] - 'a' + 'A';
            }
        }
        i++;
    }
    puts(A);
    return 0;
}
```



Q/ Verilen bir paragraf içerisindeki harflerin dağılımını bulan flowchart ve C kod.

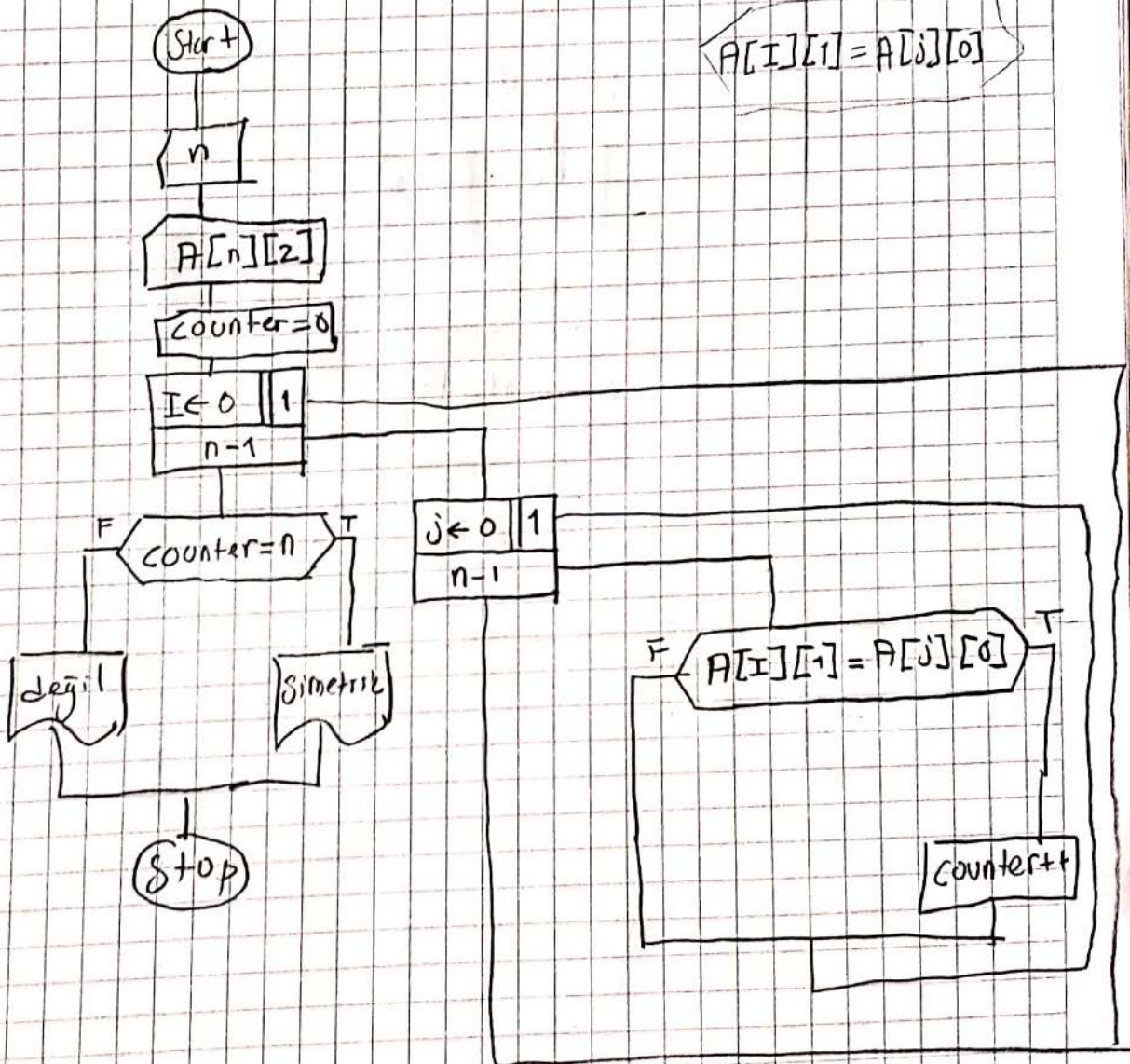


Q/

$\{(1,2), (3,2), (2,3), (1,1), (1,3), (1,4)\}$

bağıntı simetri özelliği taşıyor mu?

12
32
23
11
13
14



UYGULAMA

a=(a>b)?3:5;

```
#include<stdio.h>
int main() {
    int a,b;
    int min;
        printf("enter two positive integer");
        scanf("%d %d",a,b);
        if(a==b){
            min=a;
        }
        else{
        min=(a<b)?a:b;

    while((a%min!=0 || b%min!=0)&& min>=1){
        min--;
    }
    }
    a=a/min;
    b=b/min;
        printf("%d\n__\n%d",a,b);

    return 0;
}
```

```
#include<stdio.h>
int main() {
    int n;
    int i=2;
    int A[20];
        printf("enter number");
        scanf("%d",n);
    int k=-1;
        printf("%d=",n);

while(n!=1){
    if(n%i==0){
        k++;
        A[k]=i;
        N=N/i;
    }
    else{
        i++;
    }
}
    for(i=0; i<=k; i++){
        printf("%d",A[i]);
        if(i!=k){
            printf("x");
        }}
return 0;
}
```

```
#include <stdio.h>
#define N 100
int main() {
    int A[N];
    int i=0, n, maxIndex;
    scanf("%d", &n);
    for (i=0; i<n; i++) {
        printf("A[%d]", i);
        scanf("%d", &A[i]);
    }
    maxIndex=0;
    for(i=0; i<n; i++){
        if (A[i]>A[maxindex]) {
            maxIndex=i;
        }
    }
    int max= A[maxIndex];
    i=0;
    while (i<n) {
        if( max%A[i]==0){
            i++;
        }
        else{
        max++;
        i=0;
        }
    }
    printf("%d", max);
return 0;
}
```

```
#include <stdio.h>
int main() {
    int m[20][20];
    int i,j,n;
    printf("N=");
    scanf("%d", &n);

    m[1][0]=m[1][1]=m[0][0]=1;

    for (i=2; i<n ; i++) {
        m[i][0]=0;
        for (j=1; j<i; j++) {
            m[i][j] =m[i-1][j] + m[i-1][j-1];
        }
      m[i][i]=1;
    }
    for (i=0; i<n ; i++) {
        for (j=0 ;j<=i ; j++) {
        printf("%8d", m[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

0	0000	1010	A
1	0001	1011	B
2	0010	1100	C
3	0011	1101	D
4	0100	1110	E
5	0101	1111	F
6	0110		
7	0111		
8	1000		
9	1001		

01000111

Sınav Soruları

$(-255)_{10}$ =
00000000 11111111
11111111 00000000
\+ 1
11111111 00000001 *

(213)8 = 010 001 011

(1)

$(-11)_{10}$ =
00001011
11110100
\+ 1
11110101 *

$(144)_{10}$ = 0000000010010000

$(217)_8$ = 000000010001111

F	m	j	m
2	2	3	2
2	2	4	2
2	2	5	5
2	5	6	5
2	5	7	5
2	5	8	5
2	5	9	5

I	max	j	max
1	1	2	↑
1	1	3	↑
1	1	4	1
1	1	5	5
1	5	6	5
1	5	7	5
1	5	8	5
1	5	9	9

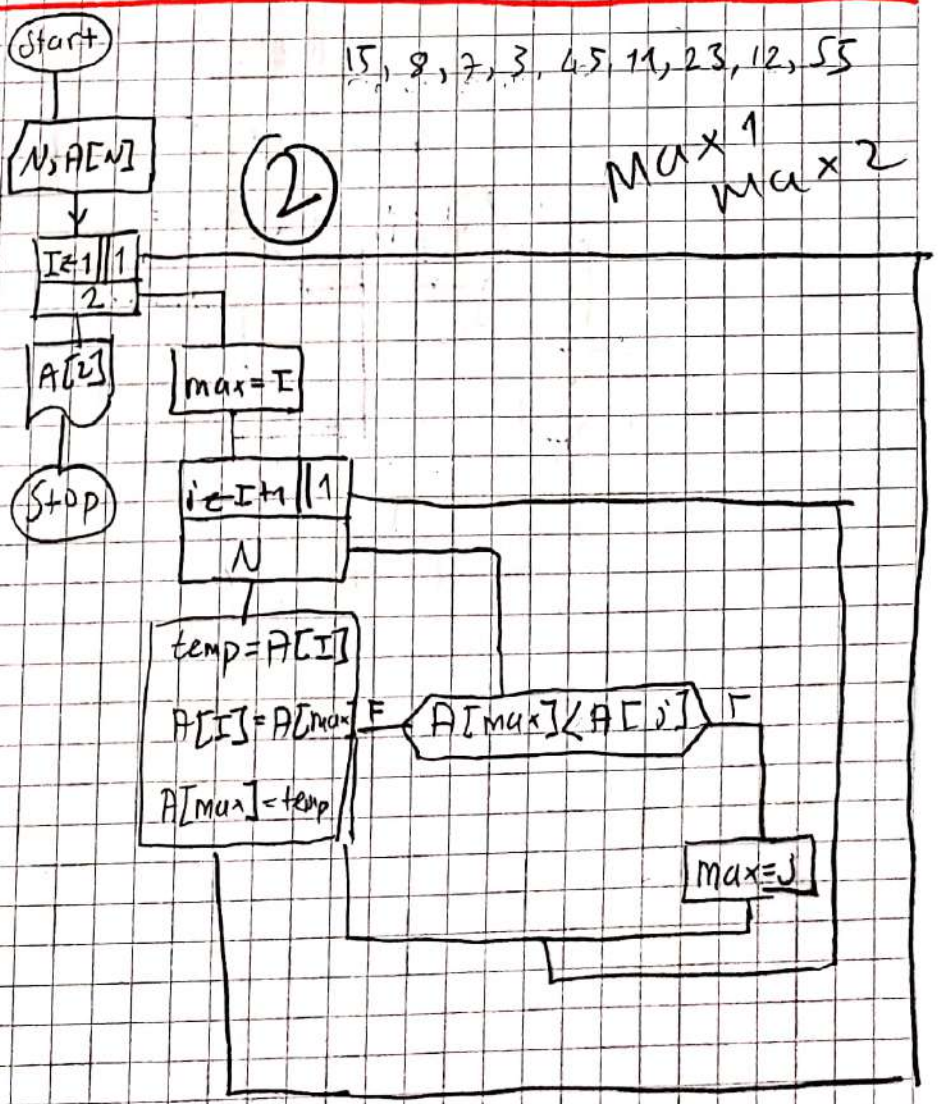
temp	A[I]	A[max]
15	55	15
8	45	8

→ A[2]

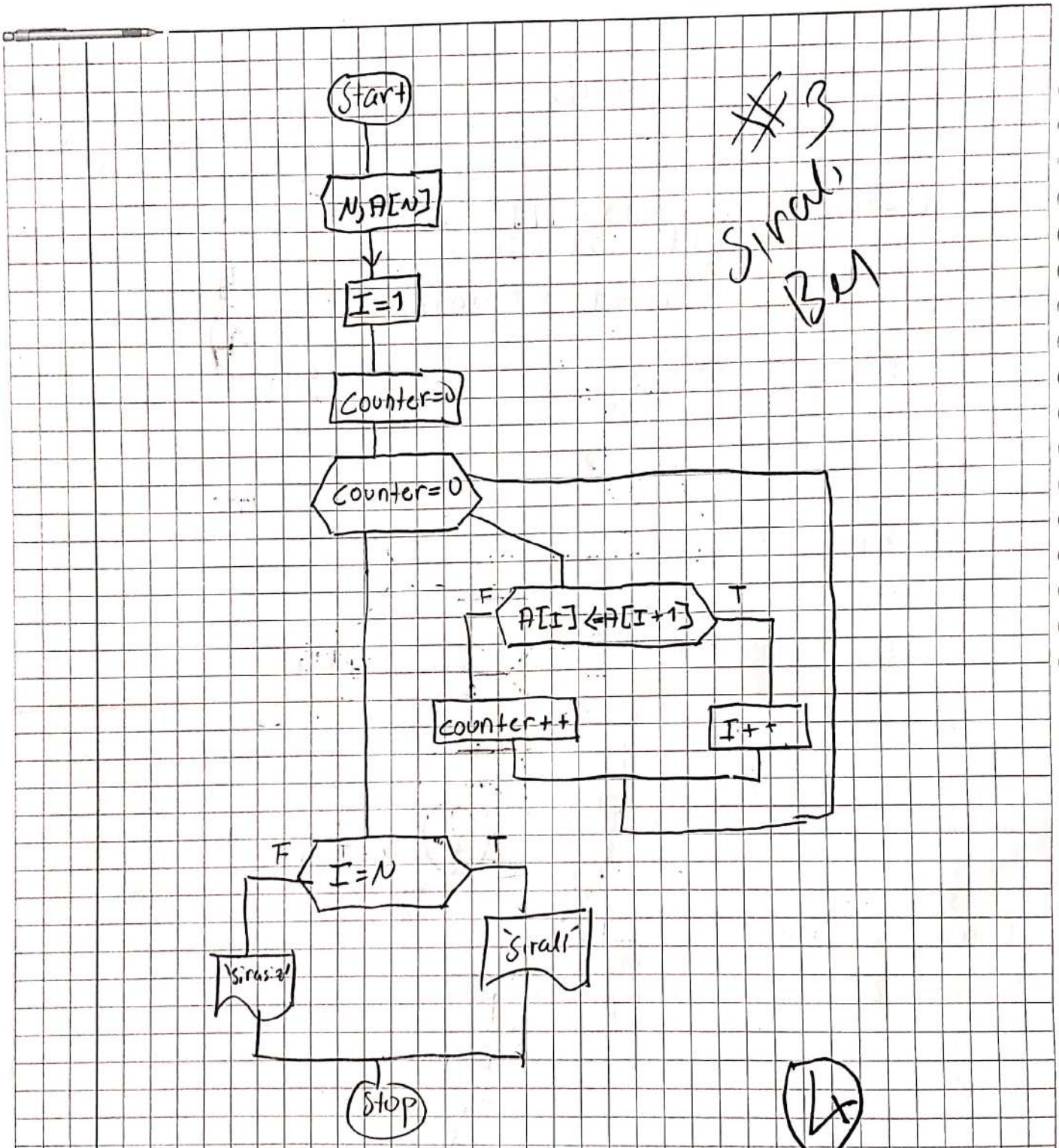
15, 8, 7, 3, 45, 11, 23, 12, 55

max 1
max 2

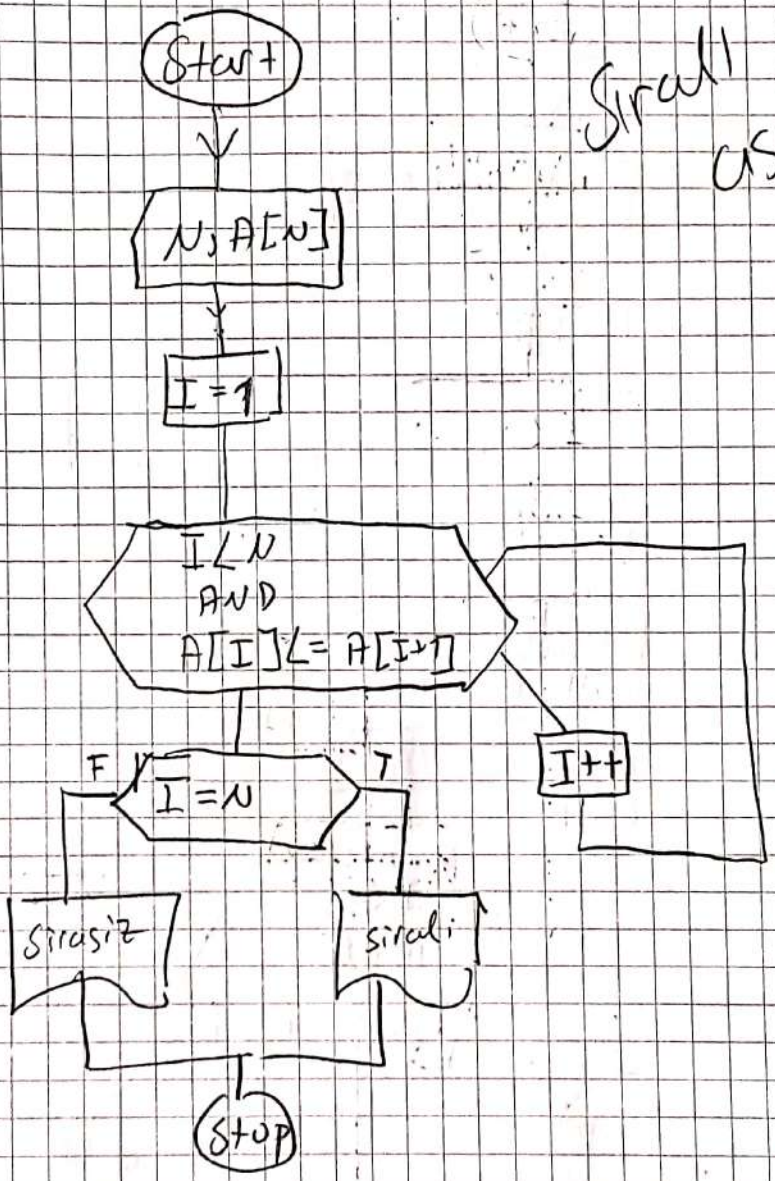
(2)



KeskinColor



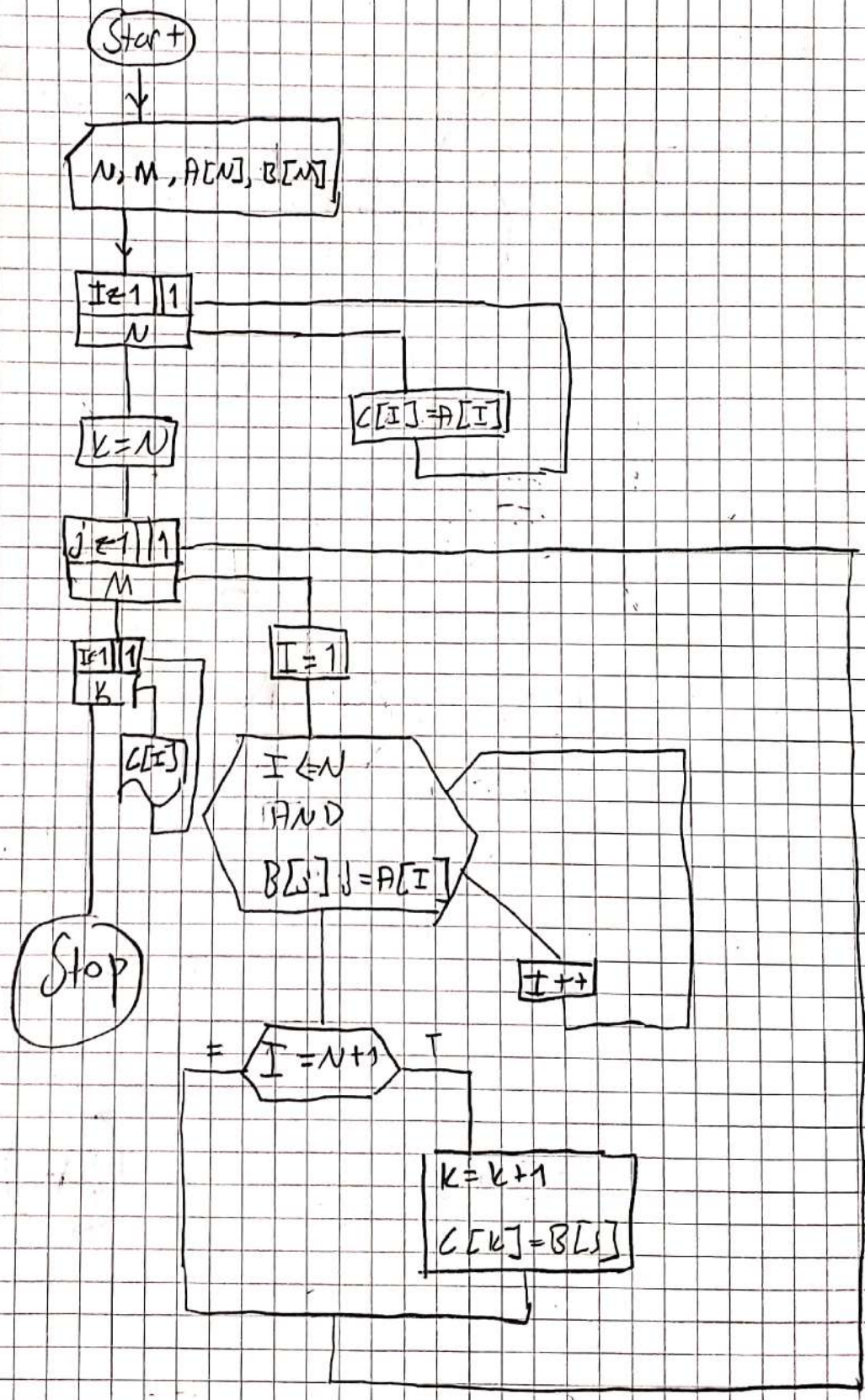
#3



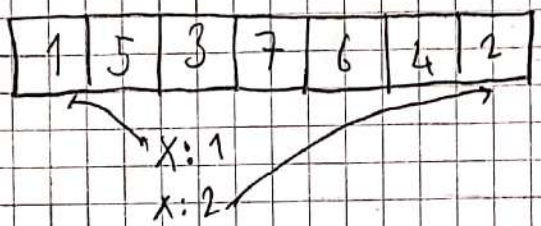
Sirali asil

4

Birleşim Kümesi

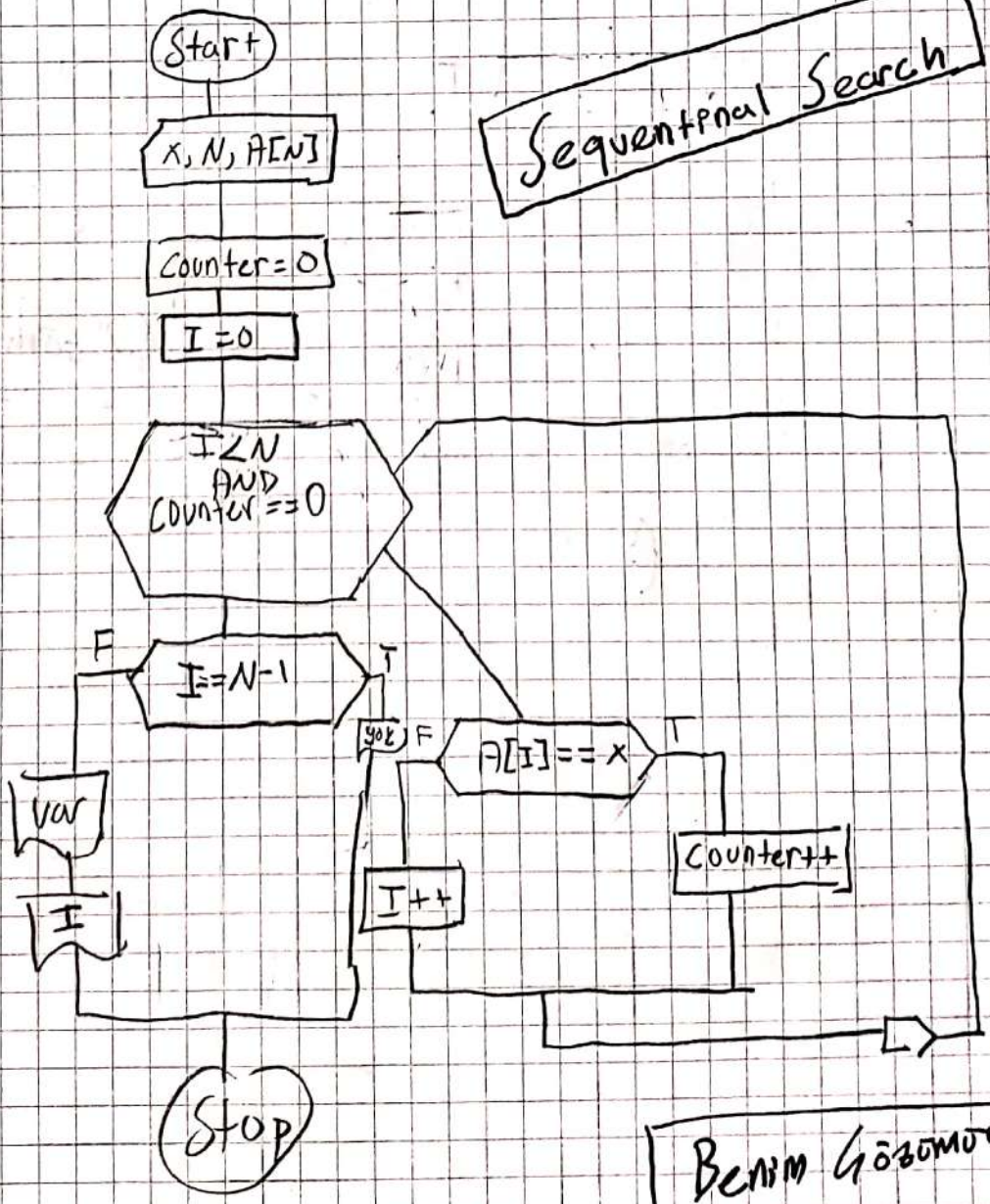


Search Algorithms

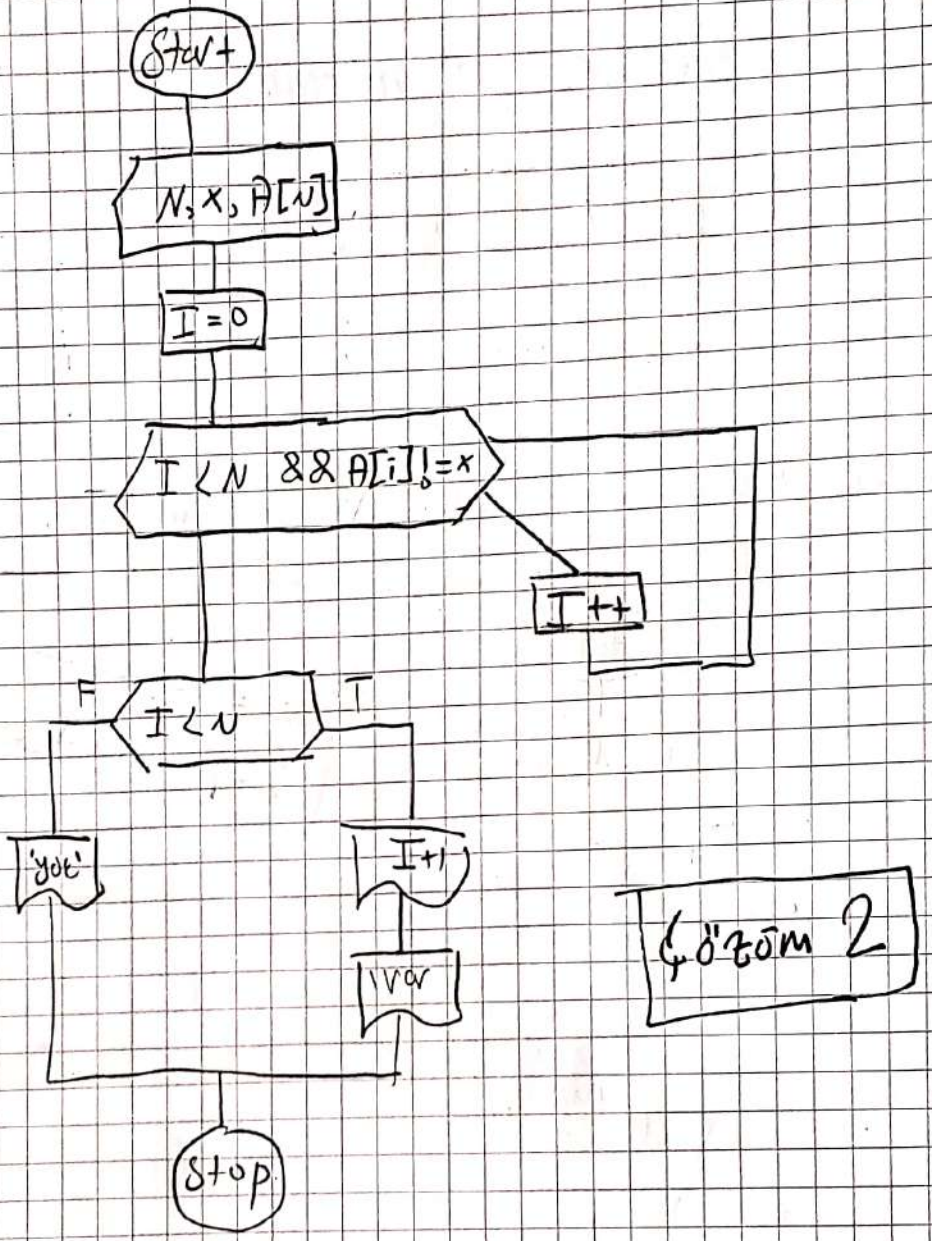


Q/ Verilen N elemanlı dizide X elemanının olup olmadığını bulan varsa kaçıncı sırada. Flowchart.

Sequential Search

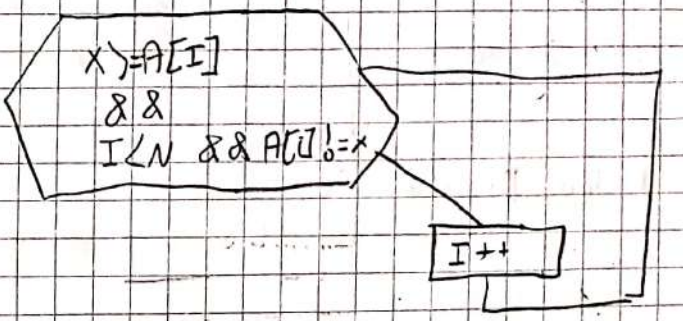


Benim Gözümün



Çözüm 2

Sorted Array



Computer Architecture

CPU ——→ Register Unit / Control Unit / Arithmetic, Logic Unit

RAM

Secondary Storage

I/O Devices

Instruction Set { RISC / CISC

Operating System

Process Management

Memory Management

File Management

I/O Management

Network Management

Job: The execution of each program

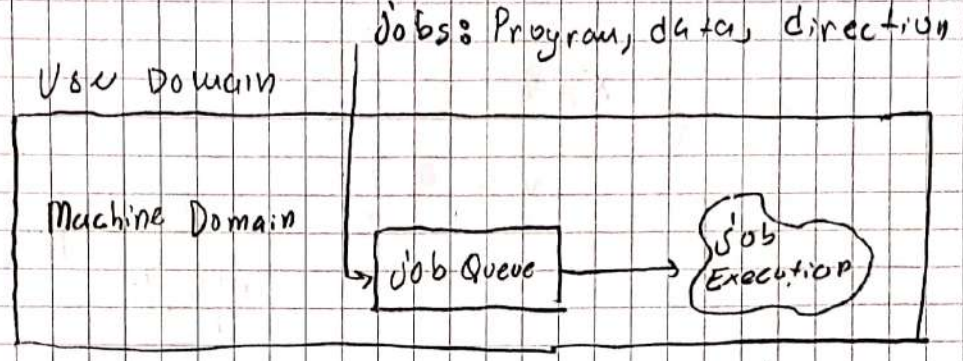
Batch processing

FIFO

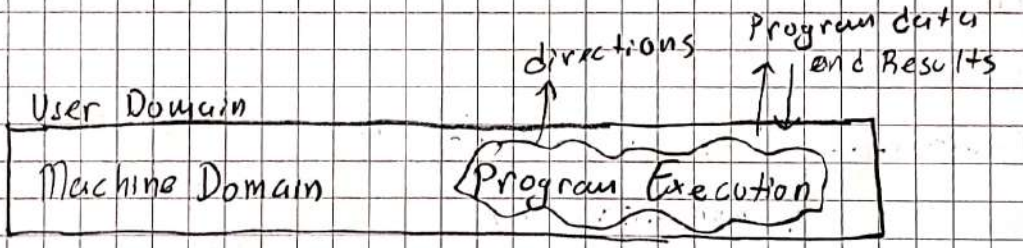
Job queue [J5 | J4 | J3 | J2 | J1]

Embeded systems

Batch Processing



Interactive Processing



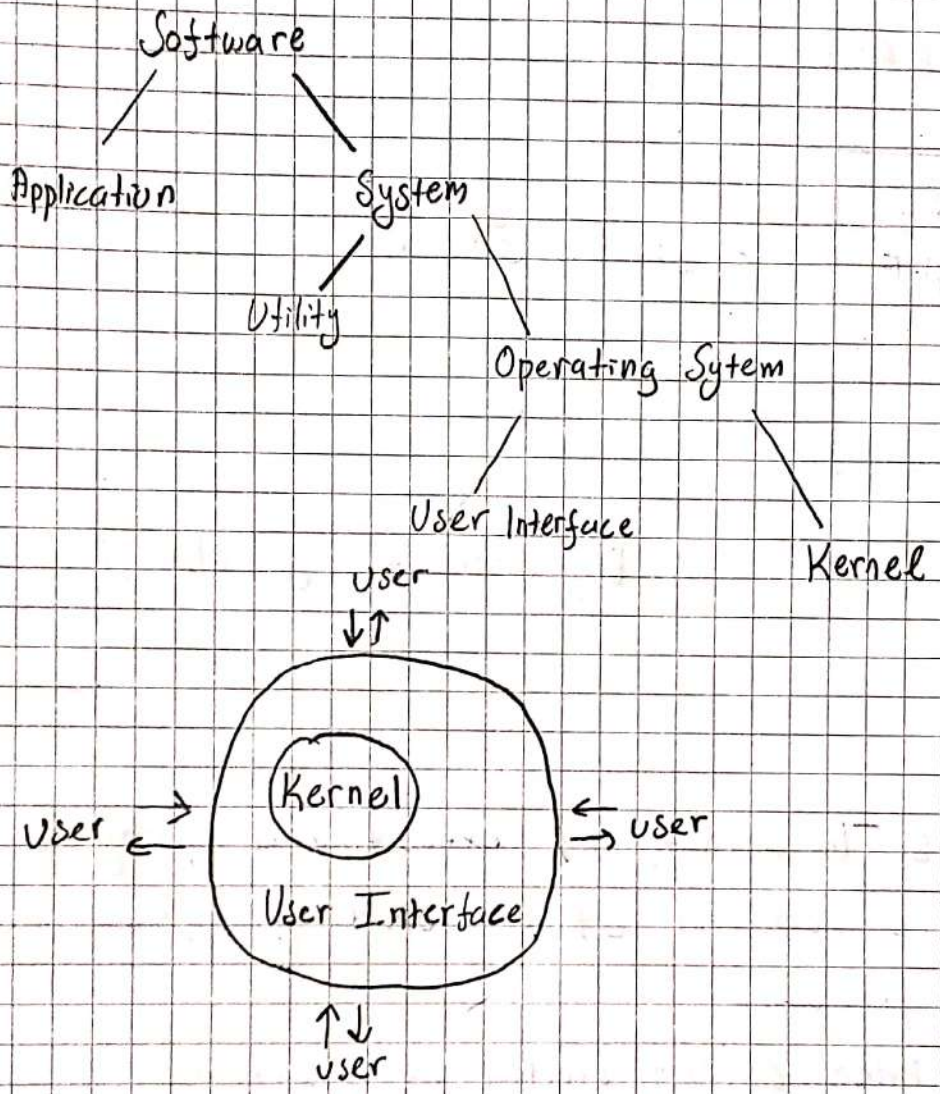
Real-Time Processing

Multi-Tasking Processing

Time Sharing

- Multi-programming
- Multi-threading [Getir dek]

Software Classification

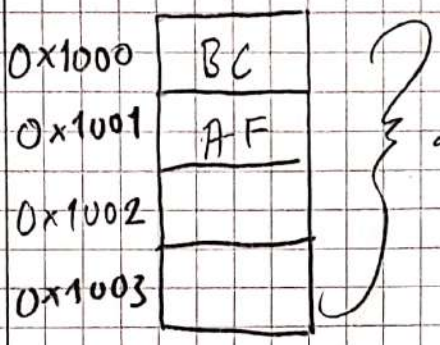


Binary Information Group Representation

Number of Bits	Common Representation
1	Bit / Digit / Flag
4	Nibble / Nybble
8	Byte / Char / Octet
16	Word
32	Long word / Double Word
64	Very long word

Little Endian / Big Endian

0xAFBC



Big Endian → Görüldüğü gibi

Process: The activity of executing a program under the control of an operating system is known as a process.

Process State: Current status of a process

Process Table: A table to keep track of all the process.

Process Switch: The procedure of changing from one process to another is called process switch

Interrupt: Each time the dispatcher awards a time slice to a process, it initiates a timer circuit that will indicate the end of the slice by generating a signal called a interrupt.

deadlock