naging I/o Functions (I/p, process, O/p)
└ essential features of computer.

2 ways to give input to pgm:

① Assigning data to variable

② I/o stmts. C I/o operations --fn. call stdio).

- Unformatted I/o stmts
- formatted I/o stmts.

① Unformatted I/o statements.

* Cannot specify the type of data.
* I/o a single/group of characters from/to I/o devices.

Input	output
getc()	putc()
getchar()	putchar()
gets()	puts()

(i) getchar() Single character input. char var = getchar();

```
char x;
x = getchar();
```

(ii) putchar() Single character output. putchar(char var);
displays one character at a time.

```
char x; putchar(x);
```

(iii) getc() accept a single character from standard input to a character variable.

```
char c = getc();
```

(iv) putc() display single character variable to std. o/p

```
putc(c);
```

getc() & putc() are used in file processing.

v) gets() & puts()

used to read the string, display/write string to o/p screen

gets(s); puts(s);

character test functions.

check the character taken as input.

isalpha(ch)	isupper(ch)
isdigit(ch)	tolower(ch)
islower(ch)	toupper(ch).

Example:

```
char x;
printf("\n Enter any alphabet");
x=getchar();
if (islower(x))
        putchar (toupper(x));
else
        putchar (tolower(x));
```

(2) Formatted I/O stmts.

Input & output arranged in a particular format

I/P	O/P
scanf()	printf()
fscanf()	fprintf()

(i) scanf()

scanf("control string", &var1, &var2 ... &varn);

control string
- character groups
- % (conversion character).

%c, %f, %d, %s

Example

getchar() & putchar()

```
Void main()
{
    char a;
    printf ("\n Enter a character");
    a = getchar();
    printf ("\n Displaying that character");
    putchar(a); putchar ('A');
}
```

gets() & puts()

```
main()
{
    char a[10];
    puts ("Enter string");
    gets (a);
    puts (a);
}
```

While loop (print n numbers).

```
# include <stdio.h>
# include <conio.h>
Void main() {
int i=1, n; clscr();
Printf ("Enter n");
Scanf ("%d", &n);
While (i<=n)
{
   printf ("%d", i);
   i++;
}
getch();
}
```

do-while loop

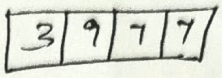
```
# include <stdio.h>
# include <conio.h>
Void main() {
int i=1, n; clscr();
printf ("Enter n");
Scanf ("%d", &n);
do {
 printf ("%d", i);
 i++;
} while (i<=n);
getch();
}
```

printf ()

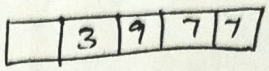
printf ("control string", var1, ... varn);

field width

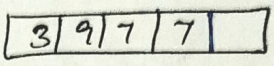
printf("%d", 3977)



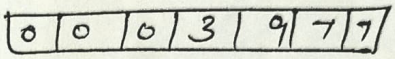
%5d, 3977



%-5d, 3977

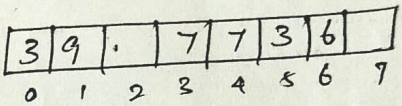


%07d, 3977

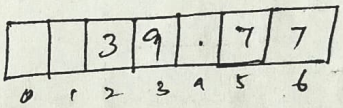


a = 39.7736

%7.4f, a



%7.2f, a



Decision Making

Control statements.

* pgm - all stmts are executed sequentially.
* No repetition cases.
* Repetition / Execution order of stmts is changed based on conditions.
  "conditional / ctrl stmts".

4 types of ctrl structures:

1. Sequential
2. Selection
3. Iteration
4. Encapsulation