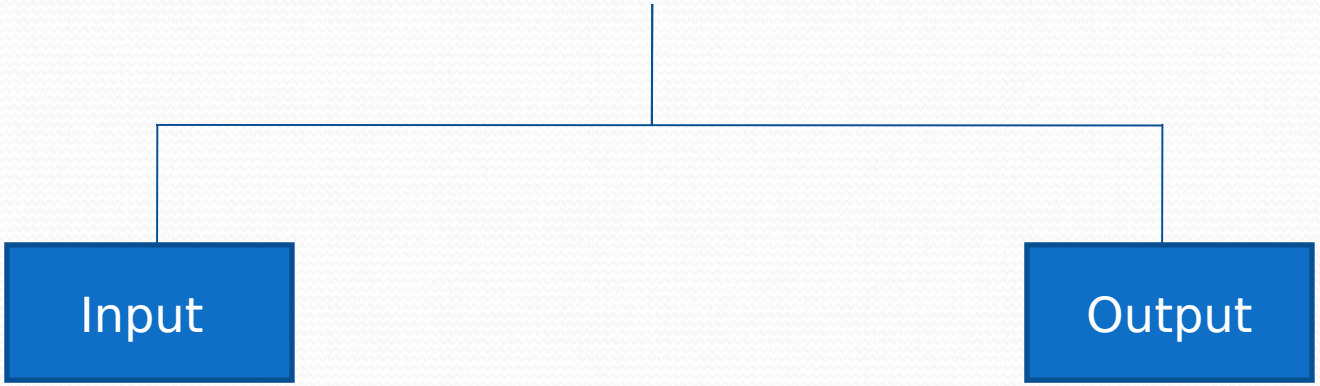
Input/Output

Types of Operations



The set of library functions that perform input-output operation is known as standard input/output library (stdio.h)

Reading a Character

- getchar();

- Accepts any character keyed in including
  - return (enter)
  - tab space

- Ex:

```
char  variable_name;
variable_name=getchar();
```

Writing a Character

- putchar(variable_name);
- Displays char represented by var_name on the terminal
- Ex:

```
char c=getchar();
putchar(c);
```

Conversion Specifications

Specifier	meaning
%c	a single character
%d or %i	decimal integer
%f or %e or %g	floating point number
%lf	long range floating point (double)
%Lf	long double
%h	short int
%s	string
%u	unsigned decimal integer
%o	octal integer
%x	hexadecimal
%[...]	Read a string of words

Formatted Input

- C provides scanf() function for entering input data
- Syntax
  - scanf(“control string”, address1, address2….);
  - Control string specifies the format in which data has to be entered
  - address1, address2 specifies the address of locations where data is to be stored

Examples Integer Numbers

- Format: %wd
  - w is the field width

- Ex 1

```
int marks;
scanf("%d",&marks);
```

- Ex 2

```
char str[30];
scanf("%s",str);
```

Value will not be stored in str

- Ex 3

```
int basic,da;
scanf("%d%d",&basic,&da);
```

- Ex 4

```
int hra,da;
scanf("%d:%d",&hra,&da);
```

- Ex 5

```
int num1,num2;
scanf("%2d %5d",&num1,num2);
```

  - 21 will be assigned to num1 and 345 will be assigned to num2 and 50 that is unread will be assigned to next scanf call

- Ex 6

```
int a,b;
scanf("%d %*d %d", &a,&b);
```

  - 123 to a
  - 456 skipped (because of *)
  - 789 to b

Examples Real Numbers

- Ex 1

  float x;

  scanf(“%f”,&x);

  - Assigns: 4.321 to x

- Ex 2

  double y;

  scanf(“%lf”,&y);

Examples char and string

- Ex 1

```
char name1[15];
scanf(“%15c”,&name1);
```

- Ex 2

```
char name2[20];
scanf(“%s”,&name2);
```

- Ex 3

```
char add[20];
scanf(“%[a-z]”,&add);
```

- Ex 4

```
char add[20];
scanf(“%[^\n]”,&add);
```

Rules for scanf

- Each variable must have a field specification
- For each field specification there must be variable address
- The scanf reads until
  - A white space is found in numeric specification
  - the maximum number of characters have been read
  - An error is detected
  - The end of file is reached

Formatted Output

- printf() is used for printing results

- printf(“control string”, arg1,arg2…..);

- Control String specifies
  - characters that will be printed on screen
  - Format Specifications
  - Escape sequence characters

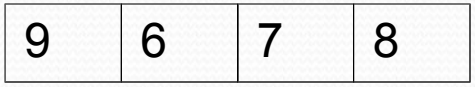
Examples

- printf(“Programming in C”);
- printf(“\n”);
- printf(“%d”,x);
- printf(“x=%d\n”,x);
- printf(“The value of a is %d”,a);

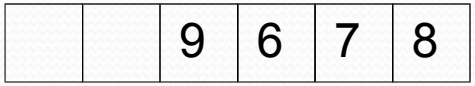
- printf does not supply new line automatically. Thus ‘\n’ is used

Integer Examples

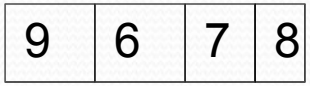
- printf(“%d”,9678);



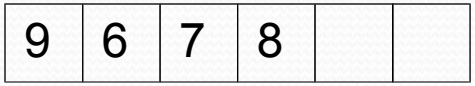
- printf(“%6d”,9678);



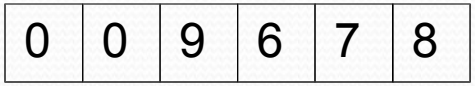
- printf(“%2d”,9678);



- printf(“%-6d”,9678);



- printf(“%06d”,9678);



Real Examples

- Syntax: %w.pf
  - w indicates the number of digits used for display
  - p indicates the number of digits to be displayed after decimal
  - Let y=98.7654;
- printf(“%7.4f”,y);

9	8	.	7	6	5	4

- printf(“%7.2f”,y);

		9	8	.	7	7

- printf(“-7.2f”,y);

9	8	.	7	7		

String Examples

- Syntax: %w.ps
  - w specifies width of field
  - p specifies only first p characters of string are displayed

- Ex:
  - char a[20]=“Hello World”;
  - printf(“%s”,a);

H	e	l	l	o		W	o	r	l	d									