

Input & Output

Er. Shiva K. Shrestha (HoD)

Department of Computer Engineering,

Khwopa College of Engineering

Data Input & Output

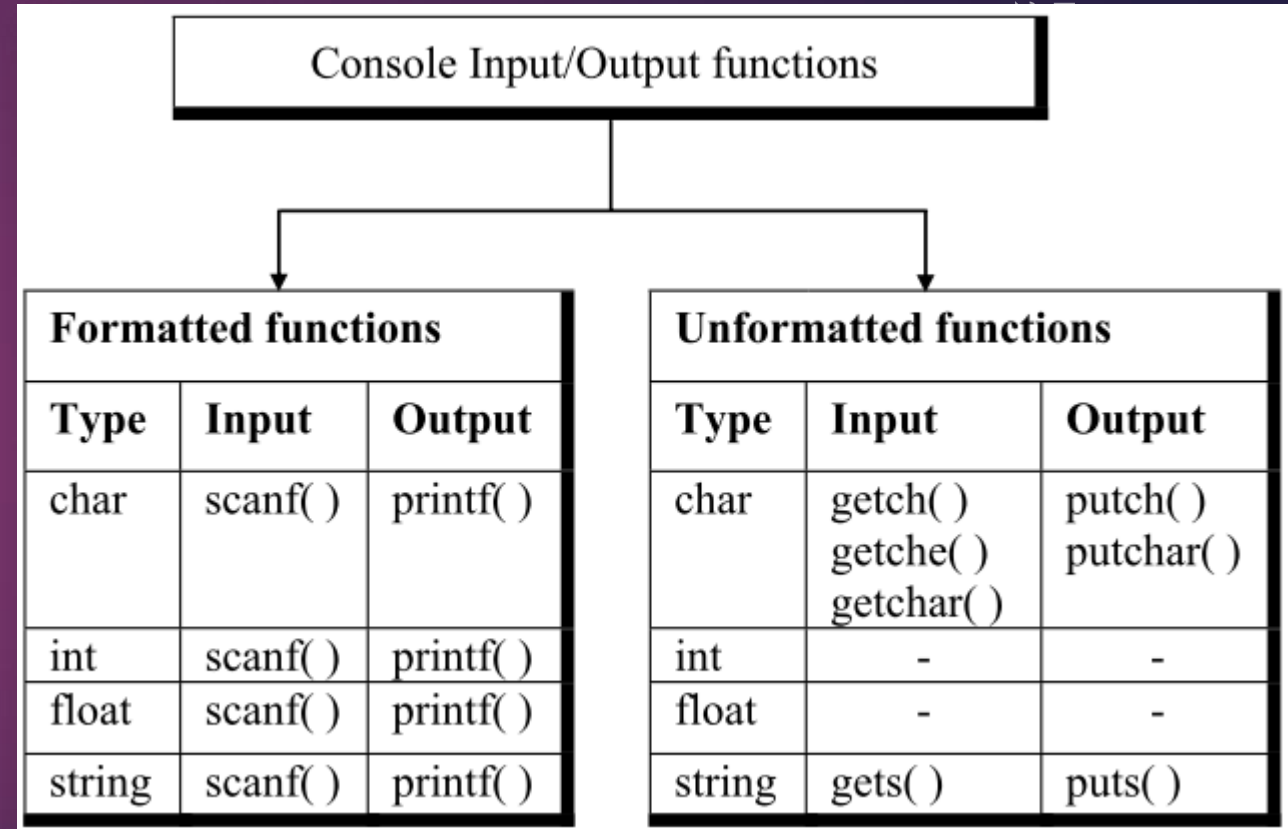
2

OUTPUT FUNCTIONS

- ▶ printf()
- ▶ putchar()
- ▶ puts()
- ▶ putch()

INPUT FUNCTIONS

- ▶ scanf()
- ▶ getchar()
- ▶ gets()
- ▶ getche()
- ▶ fgets()



Single Character Input/Output

3

Er. Shiva K. Shrestha (HoD, Com)
2019-12-24

```
IO_1.C
1 #include<stdio.h>
2 #include<conio.h>
3 void main( ){
4     char ch;
5     printf("Type one character:") ; // message to user
6     ch = getchar() ;
7     printf("The character you typed is = %c", ch) ; // output
8     getch();
9 }
```

```
IO_2.C
1 #include<stdio.h>
2 #include<conio.h>
3 void main( ){
4     char in; // character declaration of variable in.
5     printf ("Please enter one character:"); // message to user
6     in = getchar( ) ; // assign the keyboard input value to in.
7     printf("The character you typed is %c.\n", in) ; // output
8     putchar(in); // output 'in' value to standard screen.
9     getch();
10 }
```

String Input/Output

4

Er. Shiva K. Shrestha (HoD, Computer Department)
2019-12-24

```
IO_3.C
1  #include<stdio.h>
2  #include<conio.h>
3  void main( ){
4      char s[80];
5      printf("Type a string less than 80 characters: ");
6      gets(s);
7      printf("\nThe string types is: ");
8      puts(s);
9      getch();
10 }
```

DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Progra...

Type a string less than 80 characters: Artificial Intelligence

The string types is: Artificial Intelligence

Formatted Functions

5

Formatted-Input

- ▶ scanf()
- ▶ scanf ("control string", arg1, arg2, arg3, ..., argn);

Input Specifications for Integer Number: %xd

- ▶ For Long Integer: %ld
- ▶ For Short Integer: %hd

Input Specifications for Real Number:

- ▶ scanf ("%f", &variable);

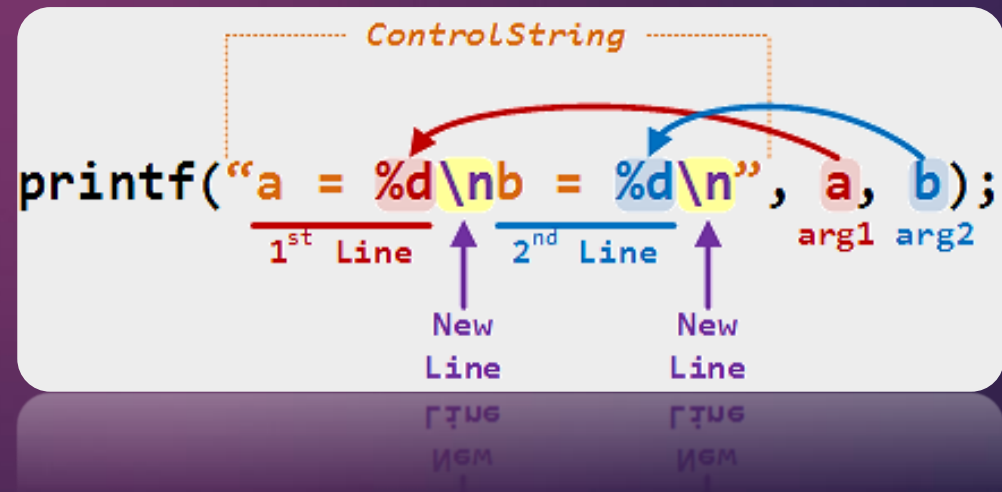
Input Specifications for a Character:

- ▶ scanf ("%c %15c", &ch, name)

Formatted-Output

- ▶ printf()
- ▶ printf ("conversion string", variable_list);

Conversion Strings and Specifiers:



Eg#4: Checking Character Type

6

```
IO_4.C
1  #include<stdio.h>
2  #include<conio.h>
3  #include<ctype.h>
4  void main(){
5      char ch;
6      clrscr();
7      printf("Press any key ...\n");
8      ch = getchar();
9      if (isalpha(ch) > 0){
10         printf("The character is a letter.");
11     }else{
12         if (isdigit (ch) > 0){
13             printf("The character is a digit.");
14         }else{
15             printf("The character is not alphanumeric.");
16         }
17     }
18     getch();
19 }
```

DOSBox 0.74, Cpu speed: max 100% cycles

Press any key ...
k
The character is a letter.
Press any key ...
2
The character is a digit.
Press any key ...
-
The character is not alphanumeric.

Eg#5: Reading & Writing of Alphabets in Reverse Case

```
#include<stdio.h>
#include <ctype.h>
void main(){
    char alphabet;
    printf("Enter an alphabet");
    putchar('\n'); /* move to next line */
    alphabet = getchar();
    if (islower(alphabet)){
        putchar(toupper(alphabet));
    }else{
        putchar(tolower(alphabet));
    }
}
```

Output

```
Enter an alphabet
a
A
Enter an alphabet
Q
q
Enter an alphabet
z
Z
```


Commonly Used printf Format Codes

8

| Specifier | Meaning |
|------------|---|
| %c | Print a character |
| %d | Print a Integer |
| %i | Print a Integer |
| %e | Print float value in exponential form. |
| %f | Print float value |
| %g | Print using %e or %f whichever is smaller |
| %o | Print octal value |
| %s | Print a string |
| %x | Print a hexadecimal integer (Unsigned) using lower case a-f |
| %X | Print a hexadecimal integer (Unsigned) using upper case A-F |
| %a | Print a unsigned integer. |
| %p | Print a pointer value |
| %hx | Print hex short |
| %lo | Print an octal long |
| %ld | Print a long integer |

Formatted Output

General Form: %[flag][field_width][.precision]conversion_char

- ▶ [flag], [field_width] & [.precision] are optional

Printing Integer Numbers

10

Note: 0 pads with zeros the leading blanks, - used for left justification

| Format | | Output | | | | | |
|----------------------------------|--|--------|---|---|---|---|---|
| <code>printf("%d",9876)</code> | | 9 | 8 | 7 | 6 | | |
| <code>printf("%6d",9876)</code> | | | | 9 | 8 | 7 | 6 |
| <code>printf("%2d",9876)</code> | | 9 | 8 | 7 | 6 | | |
| <code>printf("%-6d",9876)</code> | | 9 | 8 | 7 | 6 | | |
| <code>printf("%06d",9876)</code> | | 0 | 0 | 9 | 8 | 7 | 6 |

Printing Real Numbers: `float y = 98.7654;`

DOSBox 0.

```
98.7654
 98.77
98.77
98.765404
 9.88e+01
-9.8765e+01
9.88e+01
9.876540e+01
```

Code

```
income=1234567890.12;
printf("%-15.4e",income);
```

On display →

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|--|--|--|--|
| 1 | . | 2 | 3 | 4 | 5 | e | + | 0 | 0 | 9 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|--|--|--|--|

Left-justify
output

% - 15 . 4 e

type (m)

precision

precision

decimal point

field width

flag

signal for conversion

Format

Output

`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 | | | | | | | | | | |
|---|---|---|---|---|--|--|--|--|--|--|--|--|--|--|

`printf("%f",y)`

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|--|--|--|--|--|--|--|--|
| 9 | 8 | . | 7 | 6 | 5 | 4 | | | | | | | | |
|---|---|---|---|---|---|---|--|--|--|--|--|--|--|--|

`printf("%10.2e",y)`

| | | | | | | | | | | | | | | |
|--|--|--|---|---|---|---|---|---|---|---|--|--|--|--|
| | | | 9 | . | 8 | 8 | e | + | 0 | 1 | | | | |
|--|--|--|---|---|---|---|---|---|---|---|--|--|--|--|

`printf("%11.4e",-y)`

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|--|--|--|--|
| - | 9 | . | 8 | 7 | 6 | 5 | e | + | 0 | 1 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|--|--|--|--|

`printf("%-10.2e",y)`

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|--|--|--|--|--|--|--|
| 9 | . | 8 | 8 | e | + | 0 | 1 | | | | | | | |
|---|---|---|---|---|---|---|---|--|--|--|--|--|--|--|

`printf("%e",y)`

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|--|--|--|--|--|--|--|--|
| 9 | 8 | . | 7 | 6 | 5 | 4 | | | | | | | | |
|---|---|---|---|---|---|---|--|--|--|--|--|--|--|--|

110D Computer Department

Printing a single character

12

```
char ch = 'Y';  
putchar(ch);
```

Printing Strings

“NEW YEAR 2076”

13

Er. Shiva K. Shrestha (HoD, Computer Department)
2019-12-24

| Specification | | Output | | | | | | | | | | | | | | | | | | | |
|---------------|--|--------|---|---|--|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| %s | | N | E | W | | Y | E | A | R | | 2 | 0 | 7 | 6 | | | | | | | |
| %20s | | | | | | | | | N | E | W | | Y | E | A | R | | 2 | 0 | 7 | 6 |
| %20.10s | | | | | | | | | | | | N | E | W | | Y | E | A | R | | 2 |
| %.5s | | N | E | W | | Y | | | | | | | | | | | | | | | |
| %-20.10s | | N | E | W | | Y | E | A | R | | 2 | | | | | | | | | | |
| %5s | | N | E | W | | Y | E | A | R | | 2 | 0 | 7 | 6 | | | | | | | |

Printing Mixed Data

```
printf("%d %s %f %c", a, b, c, d);
```

► // Outputs Integer String Float Character

Formatted Input

15

General Form:

[whitespace_char][ordinary_char] %[field_width] conversion_char

- ▶ [whitespace_char], [ordinary_char] & [field_width] are optional

Points to be cared using `scanf()`

- ▶ Arguments of `scanf()` must be address except control string.
- ▶ Format specifications must match the arguments in order.
- ▶ Input data item(s) must be separated by any white space and must match the variable(s) receiving the input in the same order.
- ▶ While searching for a value, `scanf()` ignores all the white spaces like tabs, new line, etc.
- ▶ If field width is used, it should be large enough to the size of input data.
- ▶ When `scanf()` encounters an invalid value of the data item being read, the reading will be terminated.
- ▶ Any unread data item(s) in a line will be considered as a part of next `scanf()` call.

Inputting/Scanning Integer Number

17

Using Formatted Input

```
scanf("%d", &a);
```

Inputting Real Number

18

Using Formatted Input

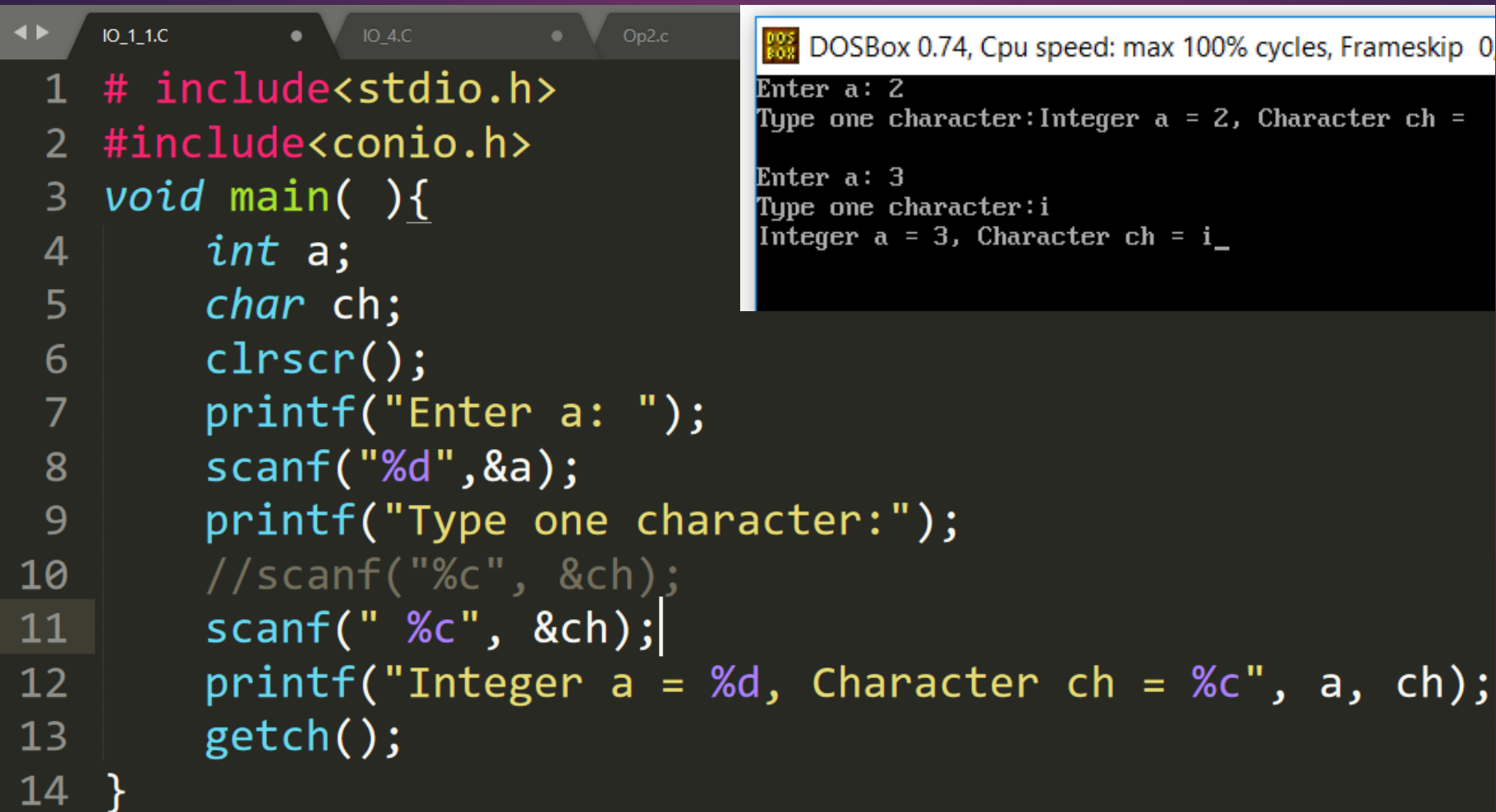
```
scanf("%f", &a);
```

Inputting a character using scanf()

19

```
char ch;
```

```
scanf("%c", &ch);
```



The image shows a code editor with three tabs: IO_1_1.C, IO_4.C, and Op2.c. The code in IO_1_1.C is as follows:

```
1 #include<stdio.h>
2 #include<conio.h>
3 void main( ){
4     int a;
5     char ch;
6     clrscr();
7     printf("Enter a: ");
8     scanf("%d",&a);
9     printf("Type one character:");
10    //scanf("%c", &ch);
11    scanf(" %c", &ch);
12    printf("Integer a = %d, Character ch = %c", a, ch);
13    getch();
14 }
```

To the right of the code editor is a DOSBox 0.74 window. The title bar reads "DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0". The terminal output shows the program's execution:

```
Enter a: 2
Type one character:Integer a = 2, Character ch =

Enter a: 3
Type one character:i
Integer a = 3, Character ch = i_
```

Inputting Strings

20

Using Formatted Input

```
char name[20];
```

```
scanf("%s", name);
```

Mixed Input

```
int a;
```

```
char b[100];
```

```
float c;
```

```
char d;
```

```
scanf("%d%s%f%c", &a, b, &c, &d);
```

► // Inputs Integer String Float Character

Interactive (conversional) programming

22



DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, F

What is your name? Sunil

Hi Sunil, Where are you from? Libali - 8, Bhaktapur

Location: Libali - 8, Bhaktapur_

```
IO_1_1.C IO_6.C IO_7.C
1  #include<stdio.h>
2  #include<conio.h>
3  void main( ){
4      char name[100], address[100];
5      clrscr();
6      printf("What is your name? ");
7      scanf("%s", name);
8      fflush(stdin);
9      printf("Hi %s, Where are you from? ", name);
10     gets(address);
11     printf("\nLocation: %s", address);
12     getch();
13 }
```


Search Set

The search set for different purposes can be defined as follows:

- ▶ To read all **decimal digits** - `%[0123456789]` or `%[0-9]`
- ▶ To read all **uppercase characters** - `%[A-Z]`
- ▶ To read all **lowercase characters** - `%[a-z]`
- ▶ To read all **decimal digits & alphabets** - `%[0-9a-zA-Z]`
- ▶ To read digits from 1 to 5 & p-z - `%[1-5p-z]`

*Note:- Minus Sign(-) used in search set must be lexically less than one after it i.e **A-Z** is valid, but **Z-A** is invalid.*

Search Set (Eg#1)

24

```
LE2_10.C x
1 #include<stdio.h>
2 #include<conio.h>
3 void main( ){
4     char str[50];
5     clrscr();
6     printf("How old are you: ");
7     scanf("%[A-Z a-z0-9]", str);
8     printf("Read string is : %s\n", str);
9     getch();
10 }
```

DOSBox 0.74, Cpu speed: max 100% cycles, Frames

How old are you: 19
Read string is : 19

How old are you: 20 years.
Read string is : 20 years

How old are you: I am 20 years old. And you?
Read string is : I am 20 years old

How old are you: Twenty years
Read string is : Twenty years

Search Set (Eg#2)

```
LE2_10.C x LE2_11.C x
1 #include<stdio.h>
2 #include<conio.h>
3 void main( ){
4     char str[50];
5     clrscr();
6     printf("Enter a string: ");
7     scanf("%[^M]", str);
8     printf("Read string is : %s\n", str);
9     getch();
10 }
```

Enter a string: Match

Read string is :

Enter a string: TestMatch

Read string is : Test

Enter a string: NPL Cricket Match

Read string is : NPL Cricket

Enter a string: 001 Sandip Lamichhane'sMatch

Read string is : 001 Sandip Lamichhane's

Enter a string: TIA Nepal - International Airport
M

Read string is : TIA Nepal - International Airport

Unformatted Functions

1. `getchar()` & `putchar()`
2. `getch()`, `getche()` & `putch()`
3. `gets()` & `puts()`

Functions `getch()`, `getche()`, `getchar()`

27

char_ip.C

```
1 #include<stdio.h>
2 #include<conio.h>
3 void main( ){
4     char ch1, ch2, ch3;
5     clrscr();
6     printf("Enter a Character: ");
7     ch1 = getch();
8     printf("\nRead character is : %c.\n", ch1);
9     ch2 = getche();
10    printf("\nRead character is : %c.\n", ch2);
11    ch3 = getchar();
12    printf("\nRead character is : %c.\n", ch3);
13    getch();
14 }
```

DOSBox 0.74, Cpu speed:

```
Enter a Character:
Read character is : a.
b
Read character is : b.
c
Read character is : c.
```

`getch()` – read instantly after pressing any key

`getche()` – read & echos instantly after pressing any key

`getchar()` – key + Enter should pressed

Q/A?

28

Thank You!

Er. Shiva K. Shrestha

computer.khwopa@gmail.com