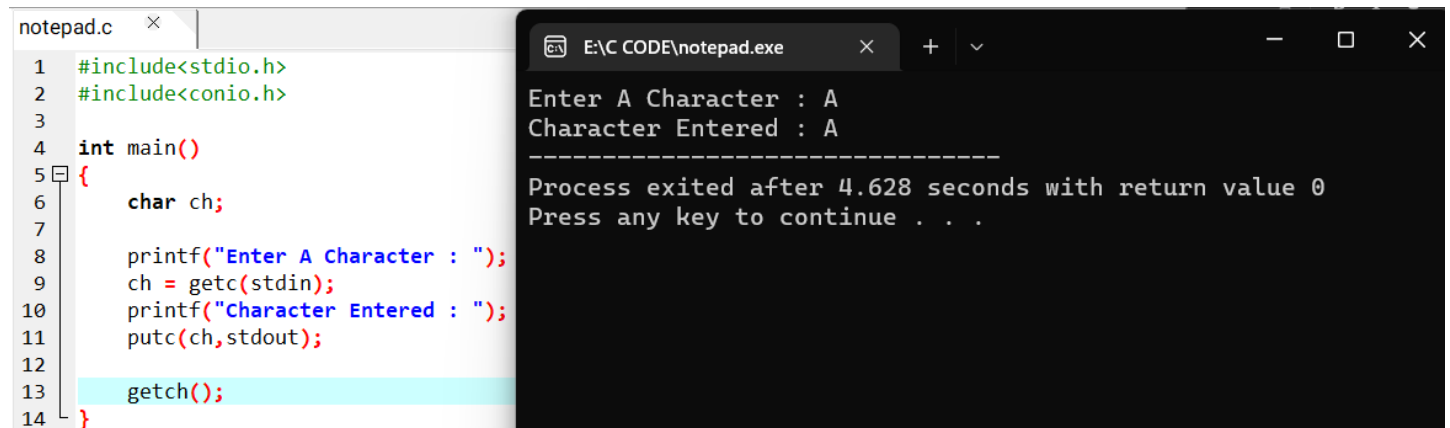


Unformatted Input Function In C Programming:

getc() function: int getc(FILE *stream)

this function gets the next character an unsigned char from the specified stream and advances the position indicator for the stream. The “stream” is the pointer to a FILE object that identifies the stream on which the operation is to be performed. It returns the character read as an unsigned char cast to an int or End Of File(EOF) or error.



The screenshot shows a C program in a text editor and its execution in a terminal. The program uses the `getc()` function to read a character from `stdin`. The code is as follows:

```
1 #include<stdio.h>
2 #include<conio.h>
3
4 int main()
5 {
6     char ch;
7
8     printf("Enter A Character : ");
9     ch = getc(stdin);
10    printf("Character Entered : ");
11    putc(ch,stdout);
12
13    getch();
14 }
```

The terminal output shows the program running, prompting for a character, receiving 'A', and displaying it. It also shows the process exit time and a prompt to press any key to continue.

getchar() function:

A `getchar()` function is a non-standard function whose meaning is already defined in the `stdin.h` header file to accept a single input from the user. In other words, it is the C library function that gets a single character (unsigned char) from the `stdin`. However, the `getchar()` function is similar to the `getc()` function, but there is a small difference between the `getchar()` and `getc()` function of the C programming language. A `getchar()` reads a single character from standard input, while a `getc()` reads a single character from any input stream.



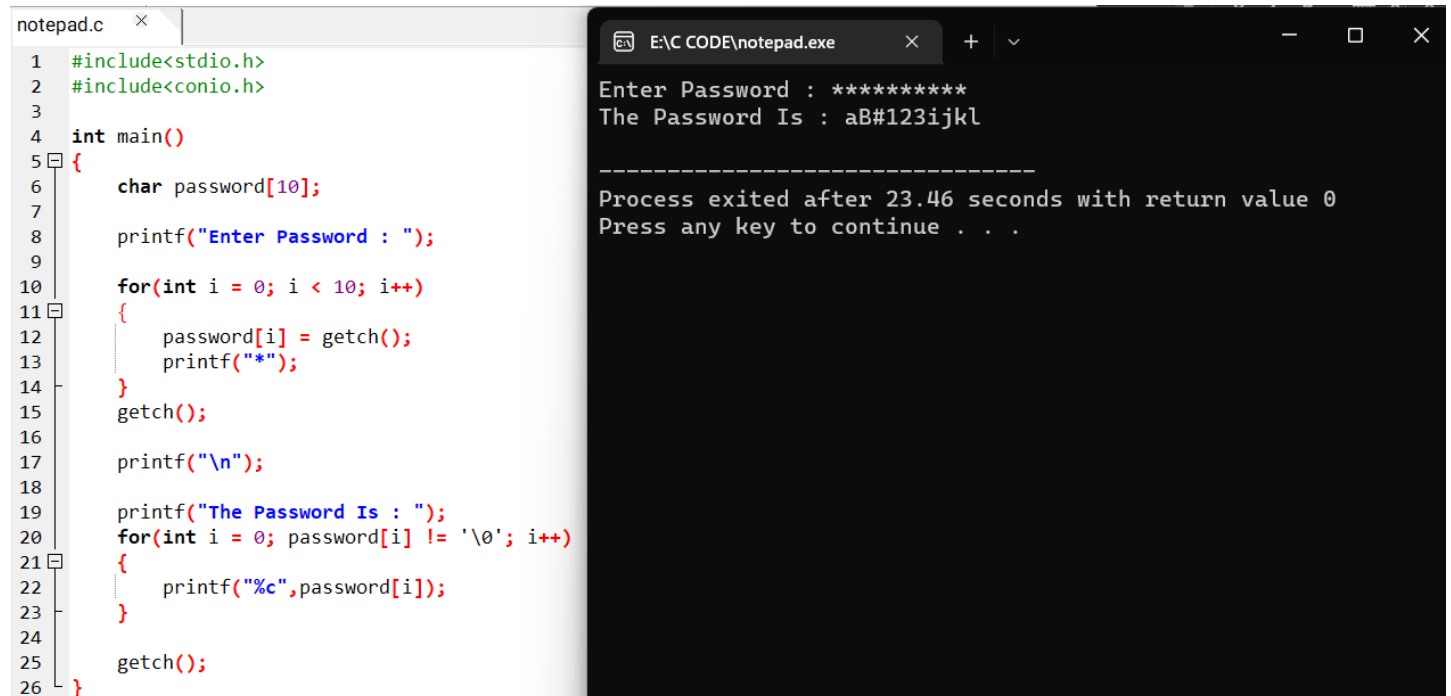
The screenshot shows a C program in a text editor and its execution in a terminal. The program uses the `getchar()` function to read a character from `stdin`. The code is as follows:

```
1 #include<stdio.h>
2 #include<conio.h>
3
4 int main()
5 {
6     char ch;
7
8     printf("Enter A Character : ");
9     ch = getchar();
10    printf("Character Entered : ");
11    putchar(ch);
12
13    getch();
14 }
```

The terminal output shows the program running, prompting for a character, receiving 'S', and displaying it. It also shows the process exit time and a prompt to press any key to continue.

getch() function:

The getch() is a predefined non-standard function that is defined in conio.h header file. We use a getch() function in a C/ C++ program to hold the output screen for some time until the user passes a key from the keyboard to exit the console screen. Using getch() function, we can hide the input character provided by the users in the ATM PIN, password, etc.



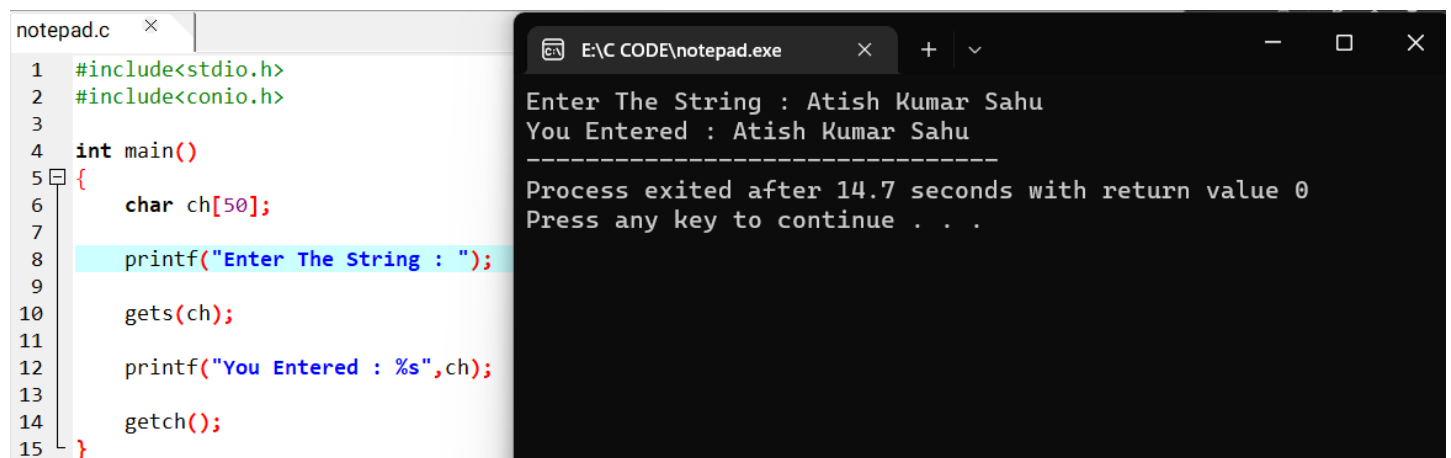
The screenshot displays a C program in a text editor and its execution in a console window. The code in notepad.c includes `stdio.h` and `conio.h`. It defines a `main` function that declares a `password` array of size 10. It prompts the user to enter a password, reads each character using `getch()` (which hides the input), and then prints the password. Finally, it uses `getch()` to pause the console before exiting.

```
1 #include<stdio.h>
2 #include<conio.h>
3
4 int main()
5 {
6     char password[10];
7
8     printf("Enter Password : ");
9
10    for(int i = 0; i < 10; i++)
11    {
12        password[i] = getch();
13        printf("*");
14    }
15    getch();
16
17    printf("\n");
18
19    printf("The Password Is : ");
20    for(int i = 0; password[i] != '\0'; i++)
21    {
22        printf("%c",password[i]);
23    }
24
25    getch();
26 }
```

The console output shows the password prompt, the password being entered as asterisks, the password being displayed, and a pause before the process exits after 23.46 seconds.

gets() function:

The gets() function enables the user to enter some characters followed by the enter key. All the characters entered by the user get stored in a character array. The null character is added to the array to make it a string. The gets() allows the user to enter the space-separated strings. It returns the string entered by the user.



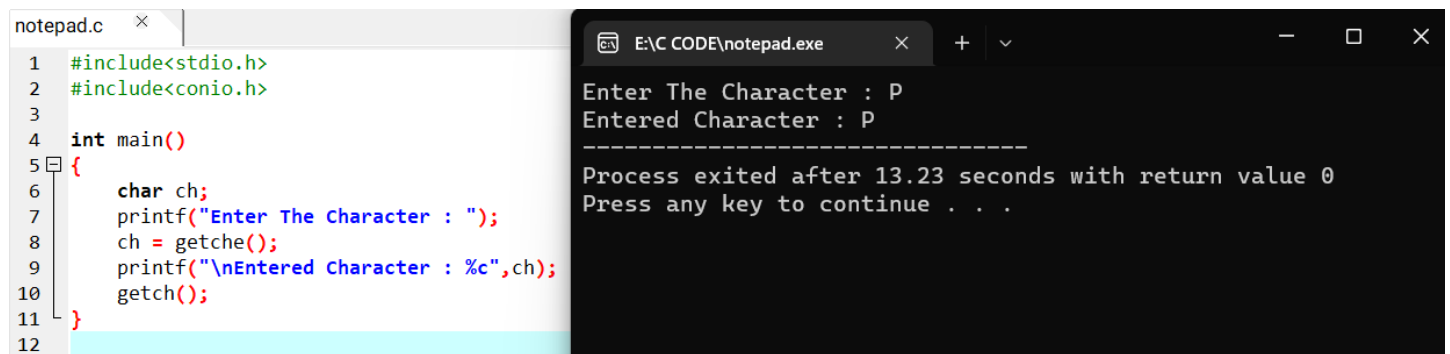
The screenshot displays a C program in a text editor and its execution in a console window. The code in notepad.c includes `stdio.h` and `conio.h`. It defines a `main` function that declares a `ch` array of size 50. It prompts the user to enter a string, reads the string using `gets(ch)`, and then prints the string. Finally, it uses `getch()` to pause the console before exiting.

```
1 #include<stdio.h>
2 #include<conio.h>
3
4 int main()
5 {
6     char ch[50];
7
8     printf("Enter The String : ");
9
10    gets(ch);
11
12    printf("You Entered : %s",ch);
13
14    getch();
15 }
```

The console output shows the string prompt, the string being entered, the string being displayed, and a pause before the process exits after 14.7 seconds.

getche() function:

getche() function is also a non-standard function and declared in "conio.h" header file. it reads a single character from the keyboard and returns it immediately without even waiting for enter key.



The screenshot displays two windows. The left window, titled 'notepad.c', shows the following C code:

```
1 #include<stdio.h>
2 #include<conio.h>
3
4 int main()
5 {
6     char ch;
7     printf("Enter The Character : ");
8     ch = getche();
9     printf("\nEntered Character : %c",ch);
10    getch();
11 }
12
```

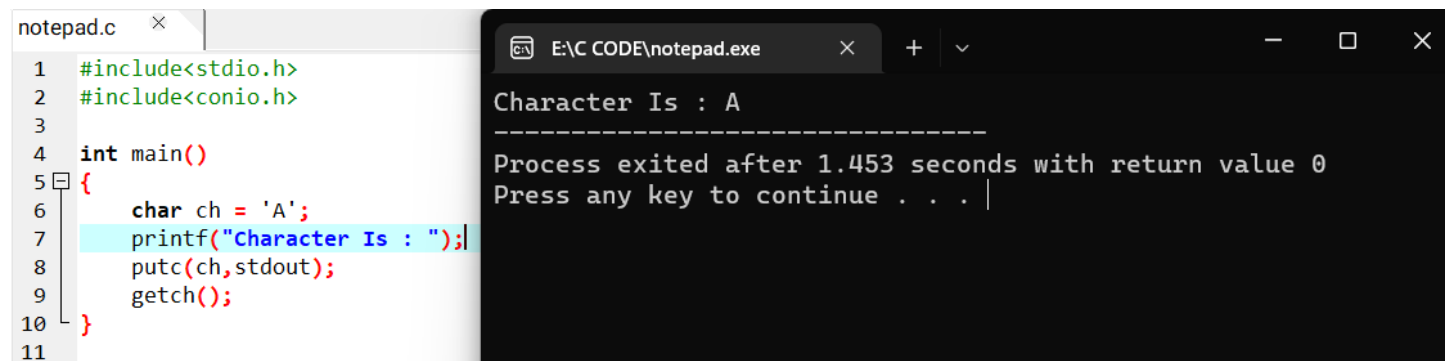
The right window, titled 'E:\C CODE\notepad.exe', shows the program's execution output:

```
Enter The Character : P
Entered Character : P
-----
Process exited after 13.23 seconds with return value 0
Press any key to continue . . .
```

53. Unformatted Output Function In C Programming:

putc() function: putc(int char, FILE *stream)

This library function writes a character (an unsigned char) specified by the argument char to the specified and advances the position indicator for the stream. "char" is the character to be written. The character is passed as its int promotion. "stream" is the pointer to a FILE object that identifies the stream where the character is to be written. This function returns the character written as an unsigned char cast to an int or End Of File(EOF) on error.



The screenshot displays two windows. The left window, titled 'notepad.c', shows the following C code:

```
1 #include<stdio.h>
2 #include<conio.h>
3
4 int main()
5 {
6     char ch = 'A';
7     printf("Character Is : ");
8     putc(ch,stdout);
9     getch();
10 }
11
```

The right window, titled 'E:\C CODE\notepad.exe', shows the program's execution output:

```
Character Is : A
-----
Process exited after 1.453 seconds with return value 0
Press any key to continue . . .
```

putchar() function: int putchar(int char)

This method in C programming is used to write a character of unsigned char type, to stdout. This character is passed as the parameter to this method. This method accepts a mandatory parameter char which is the character to be written to stdout. This function returns the character written on the stdout as an unsigned char. It also returns End Of File(EOF) when some error occurs.

The screenshot shows a C program in a text editor and its execution in a console window. The code in notepad.c includes `<stdio.h>` and `<conio.h>`, and defines a `main()` function that declares a character `ch` as 'P', prints "Character Is : ", puts the character, and then gets a key press. The console window shows the output "Character Is : P", a separator line, the message "Process exited after 6.462 seconds with return value 0", and "Press any key to continue . . .".

```
notepad.c  x
1  #include<stdio.h>
2  #include<conio.h>
3
4  int main()
5  {
6      char ch = 'P';
7      printf("Character Is : ");
8      putchar(ch);
9      getch();
10 }
11
```

```
E:\C CODE\notepad.exe  x  +  v
Character Is : P
-----
Process exited after 6.462 seconds with return value 0
Press any key to continue . . .
```

putch() function:

This function is used for printing character to a screen at current cursor location. It is unformatted character output functions. It is defined in header file "conio.h".

The screenshot shows a C program in a text editor and its execution in a console window. The code in notepad.c includes `<stdio.h>` and `<conio.h>`, and defines a `main()` function that declares a character `ch`, prints "Press An Character: ", gets a character, prints "\nCharacter Is : ", puts the character, and then gets a key press. The console window shows the output "Press An Character:", "Character Is : Q", a separator line, the message "Process exited after 3.526 seconds with return value 0", and "Press any key to continue . . .".

```
notepad.c  x
1  #include<stdio.h>
2  #include<conio.h>
3
4  int main()
5  {
6      char ch ;
7      printf("Press An Character: ");
8      ch = getch();
9      printf("\nCharacter Is : ");
10     putch(ch);
11     getch();
12 }
13
```

```
E:\C CODE\notepad.exe  x  +  v
Press An Character:
Character Is : Q
-----
Process exited after 3.526 seconds with return value 0
Press any key to continue . . .
```

puts() function:

The puts() function is very much similar to printf() function. The puts() function is used to print the string on the console which is previously read by using gets() or scanf() function. The puts() function returns an integer value representing the number of characters being printed on the console.

The screenshot shows a C program in a text editor and its execution in a console window. The code in notepad.c includes `<stdio.h>` and `<conio.h>`, and defines a `main()` function that declares a character array `ch` of size 50, prints "Enter String Value : ", gets the string, prints "The String Value : ", puts the string, and then gets a key press. The console window shows the input "Lipun Kumar Sahu", the output "The String Value : Lipun Kumar Sahu", a separator line, the message "Process exited after 7.676 seconds with return value 0", and "Press any key to continue . . .".

```
notepad.c  x
1  #include<stdio.h>
2  #include<conio.h>
3
4  int main()
5  {
6      char ch[50];
7
8      printf("Enter String Value : ");
9      gets(ch);
10     printf("The String Value : ");
11     puts(ch);
12     getch();
13 }
```

```
E:\C CODE\notepad.exe  x  +  v
Enter String Value : Lipun Kumar Sahu
The String Value : Lipun Kumar Sahu
-----
Process exited after 7.676 seconds with return value 0
Press any key to continue . . .
```