Urdu.h

A Custom C Library

By CodeMasters

Contents.

- Why Urdu.h?
- What is a Library?
- What are Functions?
- Its Use in Our Project
- List of Functions in Urdu.h
- What are Macros?
- Its Use in Our Project
- List of Macros in Urdu.h
- Output Examples
- Explanation of the Code
- Demo
- The Tools We Used

Why Urdu.h?

Standard C can be a bit intimidating for beginners, so we decided to build something unique and useful; a library that's fun to use and intuitive for Urdu speakers.

What does it do?

Urdu.h is a header-only library that:
1. It uses macros to translate standard C keywords into Roman Urdu (like *#define agr if*).

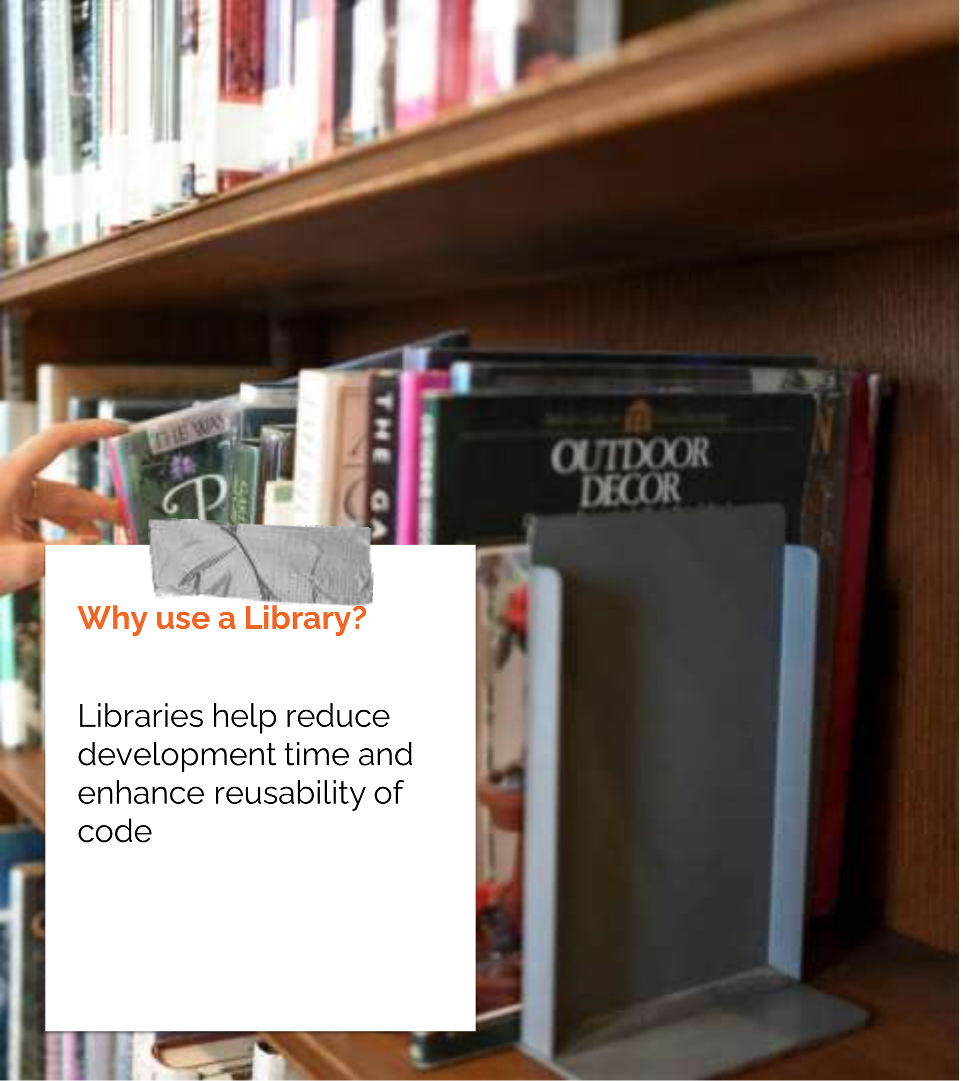
2. Simplifies pre-built functions for common tasks (e.g., *parhle* handles input in one line).

3. Removes the need to memorize format specifiers (%d, %s, etc.).

What is a Library?

- Its a pre-written collection of functions and code that you can include in your program to perform specific tasks

"*Stdio.h*" is an example of a library in C.



Why use a Library?

Libraries help reduce development time and enhance reusability of code

What are Functions?

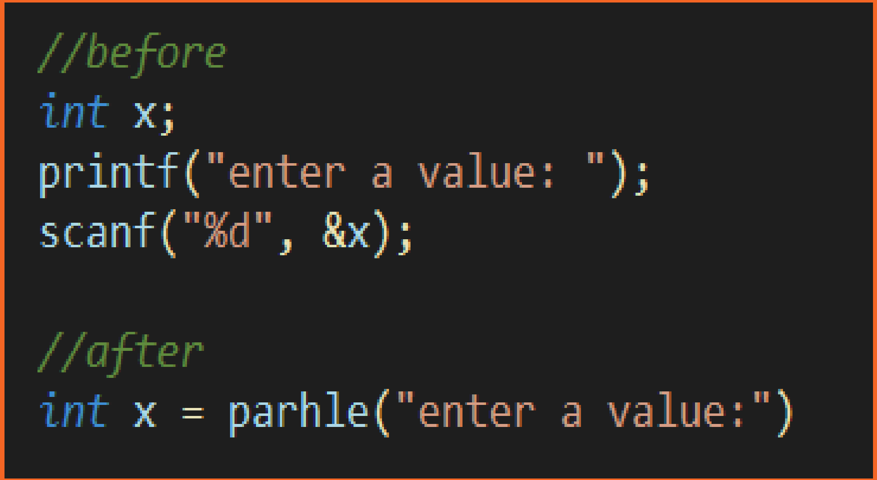
- Function is a self-contained block of code designed to perform a specific task.
- Functions are fundamental to modular programming, allowing developers to break down complex programs into smaller, more manageable, and reusable units.

Did you know?

C itself has no built-in input/output operations or other common functions, they are all defined in libraries.

Use of Functions

- We used functions to define and modify predefined functions present in C.
- The efficiency, in some cases, increased due to our modifications.



```
//before
int x;
printf("enter a value: ");
scanf("%d", &x);

//after
int x = parhle("enter a value:")
```

In C, the usual input taking would be around 3 lines of code. In Urdu.h, it reduces to just one. Thus, increasing the efficiency.

List of all the Functions

printf()	likhde()
scanf()	parhle()
puts()	likhde_line()
putchar()	likhde_harf()
strcat()	jodo()
strcpy()	nakal()
strcmp()	muqabla()
strlen()	lambai()

toupper()	bara_karo()
tolower()	chhota_karo()
isdigit()	kya_adad()
isalpha()	kya_harf()
isalnum()	kya_harfya_adad()
fopen()	file_kholo()
fclose()	file_band()
fprintf()	file_likho()

continued...

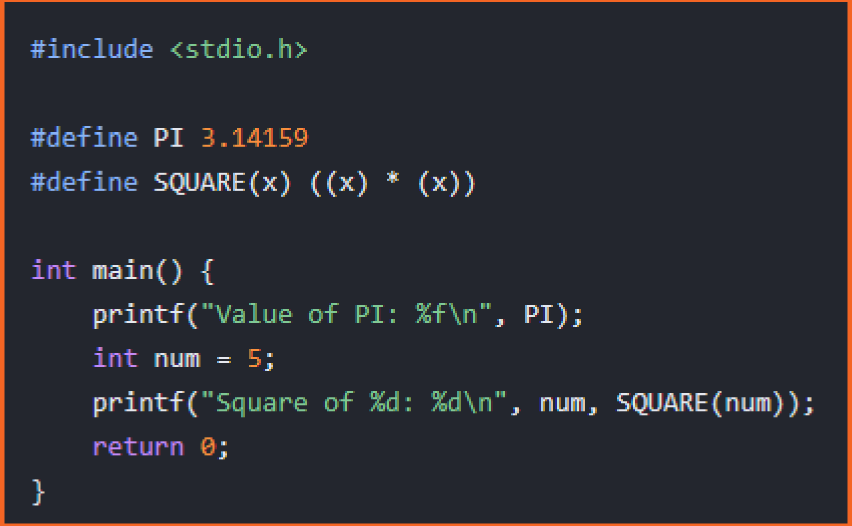
List of all the Functions

- ➔ fscanf() - file_lo()
- ➔ fgets() - line_lo()
- ➔ fputs() - line_likho()
- ➔ time() - abhi_ka_time(), abhi_ka_wakt(), aaj_ki_tareekh()
- ➔ sleep loop - ruk_jana()
- ➔ ctime() - likhde_abhi_ka_time()

What are Macros?

- Macros are symbolic names or fragments of code defined using the #define preprocessor directive.

- They are essentially text substitutions that the preprocessor performs before the actual compilation process begins.



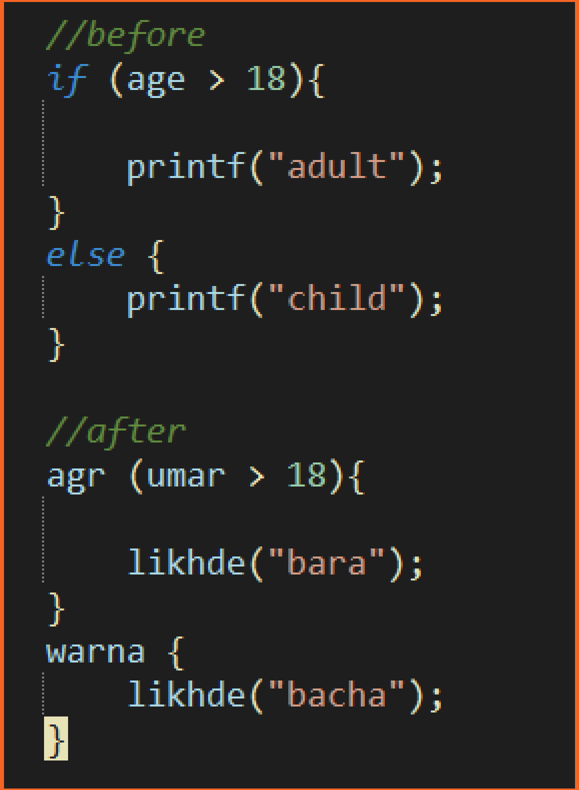
```
#include <stdio.h>

#define PI 3.14159
#define SQUARE(x) ((x) * (x))

int main() {
    printf("Value of PI: %f\n", PI);
    int num = 5;
    printf("Square of %d: %d\n", num, SQUARE(num));
    return 0;
}
```

For example, in the above program, '*PI*' is defined as 3.14159. The square of a number is also defined. Whenever the defined arguments appear in the program, they are replaced with their defined values or expressions.

Use of Macros

- We used Macros to define/change the names of certain functions C language already has.

- The use of Macros was adopted in places where functions would be inefficient.



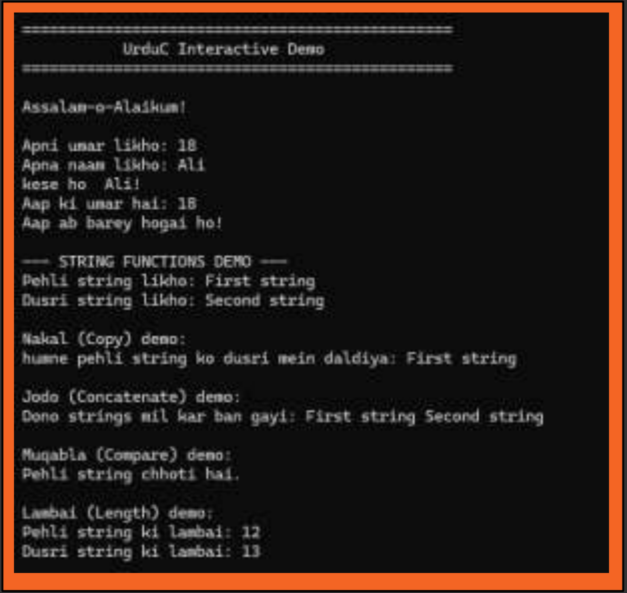
```
//before
if (age > 18){

    printf("adult");
}
else {
    printf("child");
}

//after
agr (umar > 18){

    likhde("bara");
}
warna {
    likhde("bacha");
}
```

List of all the Macros

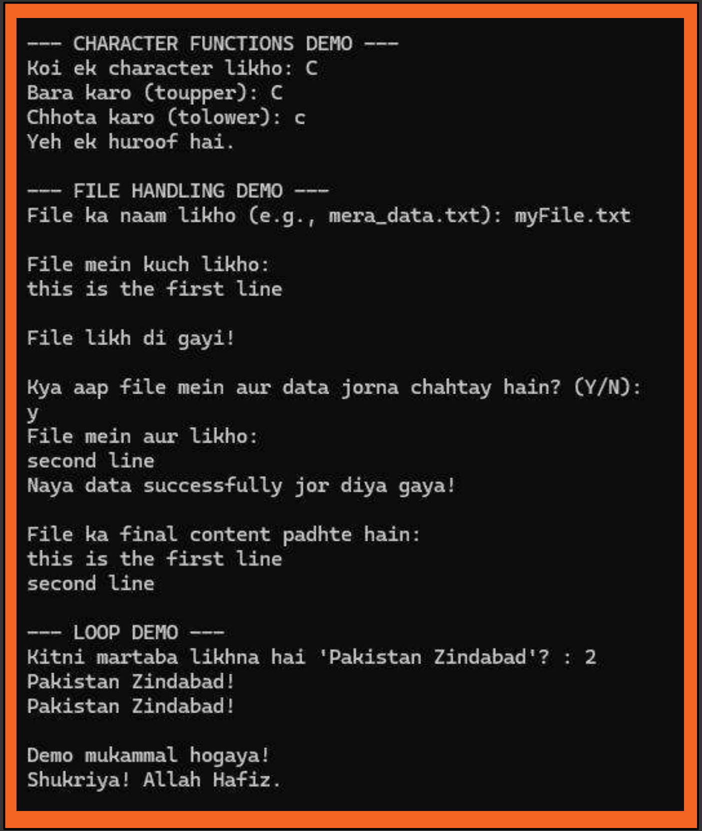
- ➔ agr - if
- ➔ warna - else
- ➔ warna_agr - else if
- ➔ badal - switch
- ➔ surat - case
- ➔ warna_sab - default
- ➔ jab_tak - while
- ➔ karo - do

- ➔ agay_chalo - continue
- ➔ ruk_jao - break
- ➔ file_kholo - fopen
- ➔ file_lo - fscanf
- ➔ bara_karo - toupper
- ➔ chota_karo - tolower

Output Examples



```
==================================================
           UrduC Interactive Demo
==================================================

Assalam-o-Alaikum!

Apni umar likho: 18
Apna naam likho: Ali
kese ho  Ali!
Aap ki umar hai: 18
Aap ab barey hogai ho!

--- STRING FUNCTIONS DEMO ---
Pehli string likho: First string
Dusri string likho: Second string

Nakal (Copy) demo:
humne pehli string ko dusri mein daldiya: First string

Jodo (Concatenate) demo:
Dono strings mil kar ban gayi: First string Second string

Muqabla (Compare) demo:
Pehli string chhoti hai.

Lambai (Length) demo:
Pehli string ki lambai: 12
Dusri string ki lambai: 13
```



```
--- CHARACTER FUNCTIONS DEMO ---
Koi ek character likho: C
Bara karo (toupper): C
Chhota karo (tolower): c
Yeh ek huroof hai.

--- FILE HANDLING DEMO ---
File ka naam likho (e.g., mera_data.txt): myFile.txt

File mein kuch likho:
this is the first line

File likh di gayi!

Kya aap file mein aur data jorna chahtay hain? (Y/N):
y
File mein aur likho:
second line
Naya data successfully jor diya gaya!

File ka final content padhte hain:
this is the first line
second line

--- LOOP DEMO ---
Kitni martaba likhna hai 'Pakistan Zindabad'? : 2
Pakistan Zindabad!
Pakistan Zindabad!

Demo mukammal hogaya!
Shukriya! Allah Hafiz.
```

The Tools We Used

These tools made our library easy to manage and code.

- ➔ **VS Code**
  To code the header file and the demo.
- ➔ **AI**
  To clarify queries on fundamental programming concepts.
- ➔ **Doxygen**
  To create a complete, professional documentation of the project.

Thank You!

Take a look at our project and **try urdu.h** yourself!