

**Bright Future Collage**  
Satdobato-15, Lalitpur



A project Report On  
**“Hotel Reservation System”**

Submitted By:  
**Prasun Ghimire**  
**Class:XII**  
**NEB Reg.no.:** \_\_\_\_\_

Under the Guidance of  
**Kamal Subedi**  
**Lecturer**  
**Department of Computer Science**  
**Bright Future Collage**

April, 2023

## **CERTIFICATE OF APPROVAL**

This is to certify that the project work entitled “**Hotel Reservation System**” in the partial fulfillment of the requirement of Computer class 12 is submitted by **Mr. Prasun Ghimire** under the supervision of Lecturer **Mr. Kamal Subedi** has been accepted.

**Lecturer:**  
**(Mr. Kamal Subedi)**  
**Department of Computer Science**  
**Bright Future College**

## **LETTER OF RECOMMENDATION**

This is to recommend that **Mr. Prasun Ghimire** of Bright Future College has done the project entitled “**Hotel Reservation System**” for the partial fulfillment of the requirement of Computer of class 12. To the best of my knowledge, this work has not been previously formed for any other degree. He has fulfilled all the requirements laid down by NEB for the submission of project work for the award of a +2 degree.

---

Examiner

## **ACKNOWLEDGMENTS**

I feel immense pleasure in acknowledging my indebtedness and heartfelt sense of gratitude to my respected Supervisor Lecturer **Mr.Kamal Subedi** sir, Department of Computer, Bright Future School/College for his sustained encouragement, regular guidance, inspiration, valuable suggestion, and great support in carrying out my project work.

I express my gratitude to the Head of the College Department **Mr.Kishor Adhikari** sir for providing us necessary materials and a well-equipped lab to learn the programming that helped in the completion of this project.

I would also like to thank my family for providing me with the financial and materials support required for the successful completion of the project.

Thank You All.

**Prasun Ghimire**

## TABLE OF CONTENT

1. Introduction of C-Language_____	1
2. History of C-Language_____	1
3. Pros&Cons of C-Language_____	1
4. Uses of C-Language_____	3
5. Branching in C-Language_____	3
6. Looping Statements in C-Language_____	4
7. Function in C-Language_____	5
8. Arrays in C-Language_____	6
9. Structures in C-Language_____	7
10. File Handling in C-Language_____	7
11. Header Files Used_____	8
12. User Defined Function used in program_____	10
13. Coding_____	12
14. Output_____	21
15. Conclusion_____	30
16. Bibliography_____	31

# C-PROGRAMMING

## ❖ INTRODUCTION

The C programming language is a procedural and general-purpose language that provides low-level access to system memory. A program written in C must be run through a C compiler to convert it into an executable that a computer can run. Many versions of Unix based operating systems (OSes) are written in C and it has been standardized as part of the Portable Operating System Interface (POSIX). Today, the C programming language runs on many different hardware platforms and OSes such as Microsoft and Linux.

## ❖ HISTORY OF C

The C programming language was developed at the former AT & T Bell Laboratories in the early 1970s by computer scientist Dennis Ritchie. The successor to the B language, C was initially developed for writing code for the Unix operating system, which at the time used assembly programs that communicated directly with the computer hardware. Assembly programs can be complex and lengthy, and programmers needed a language that promoted a user-friendly set of instructions. C fulfilled these objectives and also helped overcome the challenges that programmers experienced with BASIC, B and Basic Combined Programming Language.

Due to its popularity and flexible features, it was soon released for cross-platform usage and quickly became commercialized. C is still commonly used in web development projects and many popular languages, such as Java, PHP and JavaScript have directly or indirectly borrowed features and syntax from C.

While C has transformed over the years, it's still used commonly in lower-level programs, such as kernels. C has rapidly evolved since its inception.

## ❖ PROS AND CONS OF C

The C language comes with a set of special characteristics, making it one of the most widely used languages of all time. The following are the main benefits of using C:

- 1) **Structured.** It offers a structured programming approach for breaking down problems into smaller modules or functions that are easy to understand and modify.

- 2) **Portable.** C is machine-independent and C programs can be executed on different machines.
- 3) **Mid-level programming language.** It's a mid-level language that supports the features of both a low-level and a high-level language.
- 4) **Rich library.** It offers numerous built-in library functions that expedite the development process.
- 5) **Dynamic memory allocation.** C supports the dynamic memory allocation feature, which can be used to free the allocated memory at any time by calling the `free()` function.
- 6) **Speed.** It's a compiler-based language, which makes the compilation and execution of code faster. Since only essential and required features are included in C, it saves processing power and improves speed.
- 7) **Pointers.** C uses pointers, which improve performance by enabling direct interaction with the system memory.
- 8) **Recursion.** C enables developers to backtrack by providing code re-usability for every function.
- 9) **Extensible.** A C program can be easily extended. If code is already written, new features and functionality can be added to it with minor alterations.

C also comes with a few shortfalls, even though it's an ideal language for programming beginners due to its simple syntax, algorithms and modular structure. The following are a few disadvantages of using C:

- 1) **OOP features.** C doesn't extend its support for object-oriented programming (OOP) features, which enables the creation of sub-classes from parent classes. Unlike Java, Python or C++, multiple inheritances can't be created in C, which makes it difficult to reuse existing code.
- 2) **Name space feature.** C lacks name space features, which means the same variable name can't be reused in one scope. Without name spaces, it's impossible to declare two variables with the same name.
- 3) **Run-time checking.** C doesn't display code errors after each line of code; instead, all the errors are presented by the compiler after the program has been written. This can make code checking a challenge, especially for larger programs.
- 4) **Exception handling.** C lacks exception handling, which is the ability to handle exceptions, such as bugs and anomalies that can happen during source code.
- 5) **Constructor and Destructor.** Since C isn't object oriented, it doesn't offer constructor and destructor features. Constructing or destructing a variable in C must be done manually through a function or by other means.

**6) Garbage collection.** C isn't equipped with garbage collection. This important feature automatically reclaims memory from objects that are no longer required by the library or an app.

## ❖ USES OF C

C has a wide range of real-world applications that aren't limited to the development of OSes and applications. C is also used in areas such as graphical user interface development and integrated development environments.

The following are some use cases for the C language:

- OSes, such as Unix and all Unix applications.
- Databases, including Oracle Database, MySQL, Microsoft SQL Server and PostgreSQL, which are partially written in C.
- Language compilers, including the C compiler,
- Text editors,
- Print spoolers,
- Assemblers,
- Network drivers,
- Modern programs, such as Git and FreeBSD;
- Language interpreters, and
- Utilities, such as network drivers, mouse drivers, and keyboard drivers.

## ❖ BRANCHING IN C

Branching gets its name from the fact that the computer can select which branch to follow. Programs written in the C language execute statements one after the other. Instructions usually need to be altered in order.

A series of instructions can be rearranged in the C programming language using statements. These sentences are referred to as control statements. Moving swiftly between different program portions is made easier by these statements. Control might be transferred unconditionally or under specific restrictions.

The various types of branching statements are:

### 1. if

The if statement is a dynamic decision-making construct that can deal with a single condition or a collection of statements. These either take true or false actions. A straightforward if statement with a single block is used when a statement just has one condition.



## **2. if else**

A single condition with two distinct blocks also appears in this sentence. There are two types of blocks: true and false.

## **3. nested if**

Nested if statements refer to if statements that are contained within other if statements.

## **4. switch**

The necessity for this kind of statement with numerous alternatives or distinct situations to solve the problem simply and straight forwardly arises when there are multiple conditions present in a problem and it is quite tough to handle such a complex problem with the aid of a ladder if statement. Switch statements are used for this.

## **5. if else if**

When multiple criteria are present in a complex problem in sequential order, we can solve the issue simply by using a ladder-if or if-else-if statement.

## **❖ LOOPING STATEMENTS IN C**

Loop is used to execute the block of code several times according to the condition given in the loop. It means it executes the same code multiple times so it saves code and also helps to traverse the elements of an array.

There are 3 types of loop:

### **1. while loop**

While loop execute the code until condition is false.

**Syntax:**

```
while(condition){  
    //code  
}
```

### **2. do-while loop**

It also executes the code until condition is false. In this at least once, code is executed whether condition is true or false but this is not the case with while. While loop is executed only when the condition is true.

**Syntax:**

```
do{  
    //code  
}while(condition);
```

### 3. for loop

It also executes the code until condition is false. In this three parameters are given that is:

Initialization

Condition

Increment/Decrement

**Syntax:**

```
for(initialization;condition;increment/decrement)
{
    //code
}
```

### ❖ FUNCTIONS IN C

A function is a group of statements that together perform a task. Every C program has at least one function, which is **main()**, and all the most trivial programs can define additional functions. We can divide up our code into separate functions. How we divide up our code among different functions is up to us, but logically the division is such that each function performs a specific task.

A function **declaration** tells the compiler about a function's name, return type, and parameters. A function **definition** provides the actual body of the function. They are of two types:

#### ❖ Library Function

Library functions in C are inbuilt functions in C language. These inbuilt functions are located in some common location, and it is known as the library. All the functions are used to execute a particular operation. These library functions are generally preferred to obtain the predefined output. Also, the actions of these functions are present in their header files. For example: **stdio.h**, **conio.h**, **string.h**, **time.h**, etc.

#### ❖ User-Defined Functions

Basically a user defined function is a sub-program which groups a number of program statements into a single unit for accomplishing specific task and assigns it a name. To use user defined function, three steps are involved, they are:

- Function Prototype or Function Declaration
- Function Call
- Function Definition

The four types of user defined functions are as follows:

### **1. no argument, no return**

- It is declared as:  
void function\_name();

### **2. no argument, with return**

- It is declared as:  
return\_type function\_name();

### **3. with argument, no return**

- It is declared as:  
void function\_name(argument1,.....);

### **4. with argument, with return**

- It is declared as:  
return\_type function\_name(argument1,.....);

## **❖ ARRAY IN C**

Array in C can be defined as a method of clubbing multiple entities of similar type into a larger group. These entities or elements can be of int, float, char, or double data type or can be of user-defined data types too like structures. However, in order to be stored together in a single array, all the elements should be of the same data type. The elements are stored from left to right with the left-most index being the 0th index and the rightmost index being the (n-1) index.

Array in C are of two types:

- Single dimensional arrays, and
- Multidimensional arrays.

### **1. Single Dimensional Arrays:**

Single dimensional array or 1-D array is the simplest form of arrays that can be found in C. This type of array consists of elements of similar types and these elements can be accessed through their indices.

### **2. Multidimensional Arrays:**

The most common type of multidimensional array that is used in the C language is a 2-D array. However, the number of dimensions can be more than 2 depending upon the compiler of the user's system. These arrays consist of elements that are array themselves.

## ❖ STRUCTURES IN C

Structure is a user-defined data-type in C language which allows us to combine data of different types together. Structure helps to construct a complex data type which is more meaningful. It is somewhat similar to an Array, but an array holds data of similar type only. But structure on the other hand, can store data of any type, which is practical more useful.

## ❖ FILE HANDLING IN C

File handling refers to the method of storing data in the C program in the form of an output or input that might have been generated while running a C program in a data file, i.e., a binary file or a text file for future analysis and reference in that very program.

There are times when the output generated out of a program after its compilation and running do not serve our intended purpose. In such cases, we might want to check the program's output various times.

Now, compiling and running the very same program multiple times becomes a tedious task for any programmer. It is exactly where file handling becomes useful. The process of file handling enables a user to update, create, open, read, write, and ultimately delete the file/content in the file that exists on the C program's local file system. Here are the primary operations that we can perform on a file in a C program:

- Opening a file that already exists
- Creating a new file
- Reading content/data from the existing file
- Writing more data into the file
- Deleting the data in the file or the file altogether

## Header Files Used

### 1) **stdio.h**:

**'stdio.h'** is a header file in the C Standard Library, which is commonly used in C programming. It stands for "Standard Input/Output Header". This header file defines a set of functions and macros that provide input and output capabilities for C programs. Some of the most commonly used functions in **'stdio.h'** are **printf**, **scanf**, **fopen**, **fclose**, **fread**, **fwrite**, **gets**, and **puts**.

These functions allow a C program to interact with the standard input (usually the keyboard) and the standard output (usually the screen), as well as read from and write to files.

### 2) **conio.h**

**'conio.h'** is a non-standard but very useful header file in C. It is mostly used for console input and output functions by MS DOS compilers. It contains a list of functions, including **clrscr()**, **getch()**, **getche()**, **putch()**, **cgets()**, **cputs()**, **cscanf()**, **cprintf()**, **kbhit()**, **textcolor()**, **textbackground()**, **delline**, **gotoxy**, **wherex**, and **wherey**. They can be used to clear screen, change color of text and background, move text, check whether a key is pressed or not and to perform other tasks.

### 3) **string.h**

**'string.h'** is a header file in the C Standard Library that contains functions for manipulating strings. Some of the most commonly used functions in **'string.h'** are **strcpy**, **strcat**, **strlen**, **strcmp**, and **strchr**. These functions allow a C program to manipulate strings, which are arrays of characters.

### 4) **time.h**

**'time.h'** is a standard header file in C programming language which is used for manipulating date and time. It consists of Data Types, MACROS and Built In Functions.

The header file contains time and date function declarations to provide standardized access to time/date manipulation and formatting.

## 5) **stdlib.h**

**'stdlib.h'** is a header file and also the Standard Library of C programming language which includes functions involving memory allocation, process control, conversions and others. It has the information of memory allocation/freeing functions. Here is a table that displays some of the functions in `stdlib.h` in C language.

## User Defined Functions Used in Program

### 1. gotoxy();

**gotoxy()** is a function that was commonly used in older versions of the Turbo C/C++ compiler for DOS-based systems to set the cursor position on the screen. It was used to control the text-mode screen coordinates for outputting text or graphics at specific locations on the screen. However, it's worth noting that **gotoxy()** is a non-standard function and is not supported in modern standard C/C++.

So, I created this function as a user defined function to achieved the cursor positioning feature for modern standard C/C++ IDE (Integrated Development Environment), such as Dev-C++, Visual Studio Code, etc.

This function take the argument value i.e (x,y) coordinates value, according to the value of (x,y) the data are displayed on the screen.

### 2. report();

The function **report()** is used to display all the records of reservation. It include **costumer Id, costumer name, date of reservation, number of reservation days & room type**.

### 3. roomAvl();

The function **roomAvl()** is used to display the **total number of room** available in hotel. After reservation the **total number of room** available will be decreased according to the number of room reserved.

### 4. hotelInfo();

The function **hotelInfo()** is used to display the information of the hotel and the room types with its per day price of a room.

### 5. res();

The function **res()** is used to the get information (i.e inputs) from user and inside this another is call i.e **bill()** function.

## 6. **bill()**;

The function **bill()** is used to manipulate the data that are obtained from the customers and display the data in proper format. It displays the room type selected by the customer, total number of days entered by the customer, per day price of that type of room, number of rooms according to the number of people and finally, it display total amount and reservation id number of a customer.



# Coding

```
/*==== All the Header Files ====*/
#include <stdio.h>
#include <conio.h>
#include <string.h>
#include <stdlib.h>
#include <time.h>

/*==== No of Room ====*/
int sR = 10, kR = 20, tR = 30, totRes = 0;

/*==== All the Variables ====*/
struct Reservation
{
    char custName[30], roomType;
    int rD, rM, rY, perDay, noOfPer, resId, noOfRoom;
    float billAmt;
} sRes[90];

/*==== Function DecelARATION Part ====*/
void report();
void roomAvl();
void hotelInfo();
void res();
void bill();
void gotoxy(int x, int y);

/*==== Degrade Part ====*/

//Usage of gotoxy to set cursor position
void gotoxy(int x, int y)
{
    printf("\033[%d;%df", y, x);
}

void headerFooter()
{
    /*==== Header ====*/
    gotoxy(1, 2);
```

```

printf("+=====+");
gotoxy(1, 3);
printf("|           'Welcome To Our
Hotel'           |");
gotoxy(1, 4);
printf("+=====+");

/*==== Footer ====*/
gotoxy(1, 26);
printf("_____");
gotoxy(20, 27);
printf("Visit us at : https://www.ourhotel.com");
}

/*==== Menu ====*/
char menu()
{
    system("cls");
    headerFooter();

    gotoxy(25, 8);
    printf("+-----+");
    gotoxy(25, 9);
    printf("|   D: Data           |");
    gotoxy(25, 10);
    printf("|-----|");
    gotoxy(25, 11);
    printf("|   H: Hotel Information   |");
    gotoxy(25, 12);
    printf("|-----|");
    gotoxy(25, 13);
    printf("|   A: Availability       |");
    gotoxy(25, 14);
    printf("|-----|");
    gotoxy(25, 15);
    printf("|   R: Reservation       |");
    gotoxy(25, 16);
    printf("|-----|");

```

```

gotoxy(25, 17);
printf("|    E: Exit    |");
gotoxy(25, 18);
printf("+-----+");

gotoxy(25, 20);
printf("Enter the choice:\t");
return (getche());
}

/*=== Hotel Data ===*/
void report()
{
    int i;
    system("cls");
    headerFooter();
    if (totRes == 0)
    {
        gotoxy(25, 8);
        printf(" No Info to Display. ");
    }
    else
    {
        gotoxy(5, 6);
        for (i = 0; i < totRes; i++)
        {
            printf("\n");
            printf("\n ID   : %3d      Name: %s", sRes[i].resId,
sRes[i].custName);
            printf("\n Date: %2d/%2d   Days: %d      Room: %c \n", sRes[i].rD,
sRes[i].rM, sRes[i].perDay, sRes[i].roomType);
            printf("_____");
            printf("\n");
        }
    }
}

/*=== Room Availability ===*/
void roomAvl()
{
    system("cls");

```

```

headerFooter();
gotoxy(30, 8);
printf(" Room Information:");
gotoxy(30, 10);
printf(" Suite : Total = %d", sR);
gotoxy(30, 11);
printf(" King  : Total = %d", kR);
gotoxy(30, 12);
printf(" Travel: Total = %d", tR);
}

```

```

/*=== Hotel Information ===*/

```

```

void hotelInfo()
{
    system("cls");
    headerFooter();

    gotoxy(2, 6);
    printf(" This is 5-star Hotel which is completely Air-condition, Spacious.");
    gotoxy(2, 7);
    printf(" There are total 2 Restaurants and 1 Bar and total 90 rooms");
    gotoxy(2, 9);
    printf(" Amenities: ");
    gotoxy(15, 9);
    printf(" WIFI, Telephone, 24 Hour BAR & Restaurant and Room Service,");
    gotoxy(2, 10);
    printf(" Near By  : ");
    gotoxy(15, 10);
    printf(" Gongabu Bus Park @ 50Mtr. ");

    gotoxy(2, 13);
    printf(" Room Information:");
    gotoxy(2, 14);
    printf("+-----+-----+-----+");
    gotoxy(2, 15);
    printf("| Room Type
|      Information                               |      Price  |");
    gotoxy(2, 16);
    printf("+-----+-----+-----+");
    gotoxy(2, 17);

```

```

    printf("| Suite      | 2Rooms, 120Sq-Mtr, Sofa and King
Bed      | 10000/- |");
    gotoxy(2, 18);
    printf("+-----+-----+-----+");
    gotoxy(2, 19);
    printf("| KING      | 60Sq-Meter, Sofa, Table and King
Bed      | 5000/- |");
    gotoxy(2, 20);
    printf("+-----+-----+-----+");
    gotoxy(2, 21);
    printf("| Travel    | 40Sq-Meter, Table, Queen
Bed      | 2500/- |");
    gotoxy(2, 22);
    printf("+-----+-----+-----+");
}

```

/\*=== Reservation ===\*/

```

void res()
{
    system("cls");
    headerFooter();
    gotoxy(5, 6);
    printf(" Enter Your Name      : ");
    gotoxy(30, 6);
    gets(sRes[totRes].custName);
    // scanf("%s", &sRes[totRes].custName);
    fflush(stdin);
    gotoxy(5, 7);
    printf(" Enter Date      : dd/mm/yy");
    gotoxy(30, 7);
    scanf("%d", &sRes[totRes].rD);
    fflush(stdin);
    gotoxy(33, 7);
    scanf("%d", &sRes[totRes].rM);
    fflush(stdin);
    gotoxy(36, 7);
    scanf("%d", &sRes[totRes].rY);
    fflush(stdin);
    gotoxy(5, 8);
    printf(" Enter No of days      : ");
    gotoxy(30, 8);

```

```

scanf("%d", &sRes[totRes].perDay);
fflush(stdin);
gotoxy(5, 9);
printf(" Enter Room Type(S/K/T): ");
gotoxy(30, 9);
scanf("%c", &sRes[totRes].roomType);
fflush(stdin);
gotoxy(5, 10);
printf(" Enter No of Persons    : ");
gotoxy(30, 10);
scanf("%d", &sRes[totRes].noOfPer);
fflush(stdin);

sRes[totRes].noOfRoom = 1;

if (sRes[totRes].noOfPer > 3)
{
    sRes[totRes].noOfRoom = 0;
    sRes[totRes].noOfRoom += (sRes[totRes].noOfPer / 3);
    if (sRes[totRes].noOfPer % 3 != 0)
    {
        sRes[totRes].noOfRoom++;
    }
}
bill();
}

/*=== Bill ===*/
void bill()
{
    float damt;
    char ty[10];
    switch (sRes[totRes].roomType)
    {
        case 's':
        case 'S':
            strcpy(ty, "Suite");
            damt = 10000;
            sR = sR - sRes[totRes].noOfRoom;
            break;
        case 'k':

```

```

case 'K':
    strcpy(ty, "King");
    damt = 5000;
    kR = kR - sRes[totRes].noOfRoom;
    break;
case 't':
case 'T':
    strcpy(ty, "Travle");
    damt = 2500;
    tR = tR - sRes[totRes].noOfRoom;
    break;
}
sRes[totRes].billAmt = sRes[totRes].perDay * damt *
sRes[totRes].noOfRoom;

gotoxy(2, 13);
printf("Bill:");
gotoxy(2, 14);
printf("+-----+-----+-----+-----+-----+");
gotoxy(2, 15);
printf("|   Room Type   | Total Days | Per Day   | No of
Rooms   |   Price   |");
gotoxy(2, 16);
printf("+-----+-----+-----+-----+-----+");
gotoxy(2, 17);
printf("| %8s      |    %5d    |    %5.0f/- |    %6.0d      | %8.0f/- |",
    ty, sRes[totRes].perDay, damt, sRes[totRes].noOfRoom,
sRes[totRes].billAmt);
gotoxy(2, 18);
printf("+-----+-----+-----+-----+-----+");
gotoxy(2, 19);
printf("|   Total Bill:                                     %8.0f/- |",
    sRes[totRes].billAmt);
gotoxy(2, 20);
printf("+-----+-----+-----+-----+-----+
\t");

/*=== To Generate Random Costumer Id's ===*/
srand(time(0));

/*=== To Generate Random number between 101 to 999 ===*/

```

```

sRes[totRes].resId = rand() % (999 - 101 + 1) + 101;

gotoxy(2, 22);
printf("Your Reservation number = %d", sRes[totRes].resId);

totRes++;
}

/*==== Main Module ====*/
int main()
{
    char option;
    do
    {
        system("cls");
        option = menu();
        switch (option)
        {
            case 'd':
            case 'D':
                report();
                break;

            case 'h':
            case 'H':
                hotelInfo();
                break;

            case 'a':
            case 'A':
                roomAvl();
                break;

            case 'r':
            case 'R':
                res();
                break;

            case 'e':
            case 'E':
                gotoxy(20, 22);

```



```
        printf("Thank You ... Visit Again ... %c", 2);  
        break;  
    }  
    getch();  
} while (option != 'e' && option != 'E');  
}
```

# Output

## 1. Menu Section

```
+=====+
|                               |
|                               |
+=====+

      +-----+
      | D: Data |
      +-----+
      | H: Hotel Information |
      +-----+
      | A: Availability  |
      +-----+
      | R: Reservation   |
      +-----+
      | E: Exit          |
      +-----+

      Enter the choice:      |

-----
Visit us at : https://www.ourhotel.com
```

This is the menu section of the program consisting of 5 options which can be selected by entering 'D', 'H', 'A', 'R', 'E' for the options 'Data', 'Hotel Information', 'Availability', 'Reservation', & 'Exit' respectively.

## 2. Data Section

```
+=====+
|                                     |
+=====+

      No Info to Display. |

-----
Visit us at : https://www.ourhotel.com
```

Initially, there will be no reservation so in this section no data or information is contained to display. So, “**No Info to Display**” message will be shown.

### 3. Hotel Information Section

```
+=====+
|                                     'Welcome To Our Hotel'                                     |
+=====+

This is 5-star Hotel which is completely Air-condition, Spacious.
There are total 2 Restaurants and 1 Bar and total 90 rooms

Amenities:   WIFI, Telephone, 24 Hour BAR & Restaurant and Room Service,
Near By  :   Gongabu Bus Park @ 50Mtr.

Room Information:
+-----+-----+-----+
| Room Type | Information | Price |
+-----+-----+-----+
| Suite     | 2Rooms, 120Sq-Mtr, Sofa and King Bed | 10000/- |
+-----+-----+-----+
| KING      | 60Sq-Meter, Sofa, Table and King Bed | 5000/- |
+-----+-----+-----+
| Travel    | 40Sq-Meter, Table, Queen Bed | 2500/- |
+-----+-----+-----+

-----
Visit us at : https://www.ourhotel.com
```

In this section, the information about the hotel and the room types with its per day price of a room will be displayed.

## 4. Availability section

```
+=====+
|                               |
+=====+

Room Information:

Suite : Total = 10
King  : Total = 20
Travel: Total = 30|

-----
Visit us at : https://www.ourhotel.com
```

This section display the total number of room available in hotel. After reservation the total number of room available will be decreased according to the number of room reserved.

## 5. Reservation section

```
+=====+
|                                     |
|                                     |
+=====+

Enter Your Name      : Prasun Ghimire
Enter Date           : 04/01/2080
Enter No of days     : 2
Enter Room Type(S/K/T): S
Enter No of Persons  : 2|

-----
Visit us at : https://www.ourhotel.com
```

In this section costumer enter the data i.e costumer Id, costumer name, date of reservation, number of reservation days, room type & no of person.

## 6. Bill Section

```
+=====+
|                                     'Welcome To Our Hotel'                                     |
+=====+

Enter Your Name      : Prasun Ghimire
Enter Date           : 04/01/2080
Enter No of days     : 2
Enter Room Type(S/K/T): S
Enter No of Persons  : 2

Bill:
+-----+-----+-----+-----+-----+
| Room Type | Total Days | Per Day | No of Rooms | Price |
+-----+-----+-----+-----+-----+
| Suite    | 2          | 10000/- | 1           | 20000/- |
+-----+-----+-----+-----+-----+
| Total Bill: |           |         |             | 20000/- |
+-----+-----+-----+-----+-----+

Your Reservation number = 552|

-----
Visit us at : https://www.ourhotel.com
```

This section is used to manipulate the data that are obtained from the customers and display the data in proper format. It displays the room type selected by the customer, total number of days entered by the customer, per day price of that type of room, number of rooms according to the number of people and finally, it displays total amount and reservation id number of a customer.

According to the number of persons, the number of assigned rooms is to the customer. At most per 3 person one room is assigned to a customer.

## 7. Data Section After Reservation

```
+=====+
|                                     |
+=====+

ID   : 552      Name: Prasun Ghimire
Date: 4/ 1      Days: 2      Room: S
-----|

-----
Visit us at : https://www.ourhotel.com
```

In the section the data are displayed after the reservation. Here the customer Id, name of customer, date of reservation, total number of reservation days, & the room type.



## 8. Availability Section After Reservation

```
+=====+
|                                     |
|                                     |
+=====+

Room Information:

Suite : Total = 9
King  : Total = 20
Travel: Total = 30|

-----
Visit us at : https://www.ourhotel.com
```

After reservation the total number of room available will be decreased according to the number of room reserved.

## 9. Exit Section

```
+=====+
|                                     |
|                                     |
+=====+

      +-----+
      | D: Data |
      +-----+
      | H: Hotel Information |
      +-----+
      | A: Availability |
      +-----+
      | R: Reservation |
      +-----+
      | E: Exit |
      +-----+

      Enter the choice:      e

      Thank You ... Visit Again ... █

-----
Visit us at : https://www.ourhotel.com
```

When customer pressed 'e' or 'E' key the program will be end.

## **Conclusion**

To use the system, customers must complete a hotel reservation form that captures their specific requirements. Once the form is filled out, the system assigns a room to the customer based on their stated preferences and the details provided in the form. Customers have the ability to select a room that meets their specific needs, such as room size, amenities, location, and other important features.

In addition to reserving a room, customers can also access important information about the hotel, such as the number of available rooms and other details about the facilities and services offered. This provides customers with a comprehensive understanding of what the hotel has to offer.

After selecting a room, customers are prompted to enter the number of days they intend to stay. The system calculates the total cost of the reservation based on the number of days selected and any other relevant fees. Customers are then given the option to either confirm or cancel the reservation.

Once the customer has completed the reservation process, they can view their reservation details in the reservation list. This allows customers to review their reservation details, such as the selected room type, check-in and check-out dates, and the total cost.

Overall, this hotel reservation system provides customers with a seamless and efficient way to book a room that meets their individual preferences and scheduling requirements. By offering essential information about the hotel and allowing customers to review their reservation details, this system ensures a hassle-free booking experience.

## **Bibliography**

1. “Asmita Essentials of Computer Science”
2. <https://chat.openai.com/>
3. <https://youtube.com/>
4. <https://www.google.com/>
5. <https://www.bing.com/search?q=Bing+AI&showconv=1>
6. <https://stackoverflow.com/>
7. <https://www.w3schools.com/c/index.php>
8. <https://www.programiz.com/c-programming>