Mini Project – Vaccine Registration System

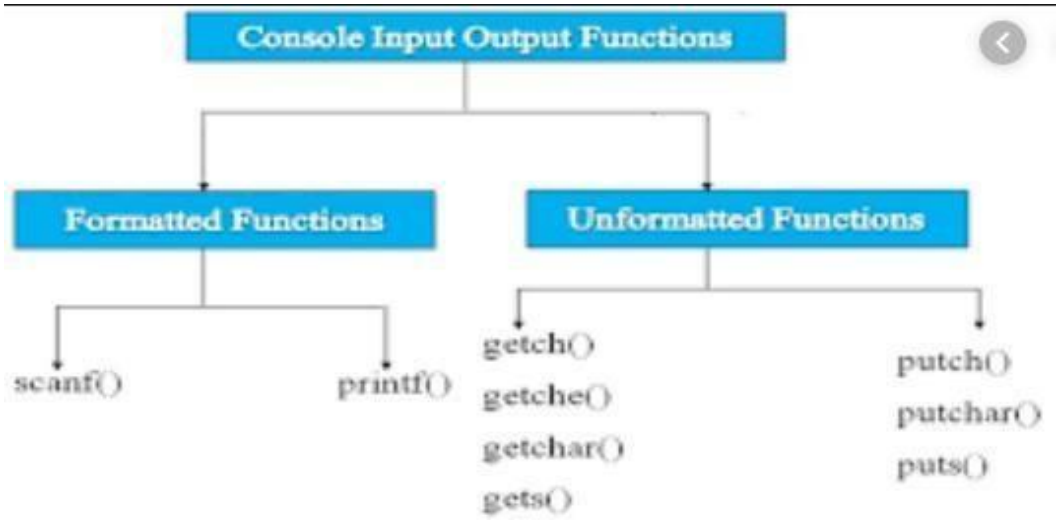
Aim:

To develop applications for the given real world applications in C.

Concepts Involved:

Input / Output Statements

- We know that input, process, output are the three essential features of computer program. The program takes some input data, processes it and gives the output.
- We have two methods for providing data to the program.
  - Assigning the data to the variables in a program
  - By using the input/output statements.
- Two types of I/O statements are available in 'C' language and all input and output operations are carried out through the function call. Several functions are available for input/output operations in 'C'. These functions are collectively known as the standard I/O library.
  - Unformatted Input / Output statements
  - Formatted Input / Output statements.



Decision Making Statements

- In decision control statements (if-else and nested if), group of statements are executed when condition is true. If condition is false, then else part statements are executed.
- There are 3 types of decision making control statements in C language. They are,

  1. if statements

  2. if else statements

  3. nested if statements

Decision control statements	Syntax/Description
if	Syntax: if (condition) { Statements; }Description: In these type of statements, if condition is true, then respective block of code is executed.

if...else	Syntax:
	if (condition)
	{ Statement1; Statement2; }
	else
	{ Statement3; Statement4; }Description:
	In these type of statements, group of statements are executed when condition is true. If condition is false, then else part statements are executed.

nested if	Syntax:
	if (condition1){ Statement1; }
	else_if(condition2)
	{ Statement2; }
	else Statement 3;Description:
	If condition 1 is false, then condition 2 is checked and statements are executed if it is true. If condition 2 also gets failure, then else part is executed.

Looping Statements

- Loop control statements in C are used to perform looping operations until the given condition is true.
- Control comes out of the loop statements once condition becomes false.
- There are 3 types of loop control statements in C language. They are,
    1. for
    2. while
    3. do-while

Syntax for each C loop control statements are given in below table with description

Loop Name	Syntax
for	for (exp1; exp2; expr3) { statements; }Where, exp1 – variable initialization (Example: i=0, j=2, k=3) exp2 – condition checking (Example: i>5, j<3, k=3) exp3 – increment/decrement (Example: ++i, j–, ++k)
while	while (condition) { statements; }where, condition might be a>5, i<10
do while	do { statements; } while (condition);where, condition might be a>5, i<10

Case Control Statements:

- The statements which are used to execute only specific block of statements in a series of blocks are called case control statements.
- There are 4 types of case control statements in C language. They are,
  1. switch
  2. break
  3. continue
  4. goto

1.Switch case statement

- Switch case statements are used to execute only specific case statements based on the switchexpression.

Syntax:

```
switch (expression)
{
   case label1:  statements;
                 break;
   case label2:  statements;
                 break;
   case label3:  statements;
                 break;
   default:      statements;
                 break;
}
```

2.Break statement

- Break statement is used to terminate the while loops, switch case loops and for loops from the subsequent execution.

  Syntax: break;

3.continue statement

- Continue statement is used to continue the next iteration of for loop, while loop and do-while loops. So, the remaining statements are skipped within the loop for that particular iteration.

  Syntax : continue;

4.goto statement

- goto statements is used to transfer the normal flow of a program to the specified label in the program.

  Syntax:

```
{
      .......
      go to label;
      .......
      .......
      LABEL:
      statements;
}
```

Strings:

- C Strings are nothing but array of characters ended with null character („\0").
- This null character indicates the end of the string.
- Strings are always enclosed by double quotes. Whereas, character is enclosed by single quotes in C.

String Declaration

Method 1:

```
char address[]={'W', 'O', 'R', 'K', 'S', '\0'};
```

Method 2: **The above string can also be defined as –**

```
char address[]="WORKS";
```

In the above declaration NULL character (\0) will automatically be inserted at the end of the string.

NULL Char „\0"

'\0' represents the end of the string. It is also referred as String terminator & Null Character.

Modular Programming

If the program is divided into number of functional parts, then we use to call it as modular programming.

If the main program is divided into sub programs, then we can independently code each sub module later combine into single unit. This type of individual modules is termed as functions.

Functions:

C functions are basic building blocks in a program. All C programs are written using functions to improve re-usability, understandability and to keep track on them.

A large C program is divided into basic building blocks called C function. C function contains set of instructions enclosed by "{ }" which performs specific operation in a C program. Actually, Collection of these functions creates a C program.

Library Function Vs User Defined Function

S.No	Library Function (LF)	User Defined Function (UDF)
1	LFs are predefined function.	UDFs are the function which is created by user as per his / her own requirements.
2	LFs are part of header file (Such as math.h) which is called runtime.	UDFs are part of the program whichcompile runtime.
3	In LF, the name of function is given by developers.	In UDF, the name of function is decided by user.
4	In LF, name of function cannot be changed.	In UDF, name of function can be changedany time.
5	Example: sin, cos, power and etc.,	Example: fibo, addition, and etc.,

Function Declaration, Function Call and Function Definition

There are 3 aspects in each C function. They are,

- Function declaration or prototype – This informs compiler about the function name, function parameters and return value‟s data type.
- Function call – This calls the actual function
- Function definition – This contains all the statements to be executed.

C functions aspects	syntax
function definition	Return_type function_name (arguments list) { Body of function; }
function call	function_name (arguments list);
function declaration	return_type function_name (argument list);

Structure:

Structure is a collection of different data types which are grouped together and each element in a structure is called member.

- If you want to access structure members in C, structure variable should be declared.
- Many structure variables can be declared for same structure and memory will be allocated for each separately.
- It is a best practice to initialize a structure to null while declaring, if we don‟t assign any values to structure members.

Syntax	struct student
	{
	int a;
	char b[10];
	}

Example	a = 10; b = “Hello”;

Pointer:

A pointer is a variable that stores the address of another variable. Unlike other variables that hold values of a certain type, pointer holds the address of a variable. For example, an integer variable holds (or you can say stores) an integer value, however an integer pointer holds the address of an integer variable.

Declaring a pointer

Like variables, pointers have to be declared before they can be used in your program.

A pointer declaration has the following form.

date_type *pointer_variable_name;

where,

- **data_type** indicates the type of the variable that the pointer points to.
- The asterisk (*: the same asterisk used for multiplication) which is indirection operator, declares a pointer.

File Handling:

File is a collection of bytes that is stored on secondary storage devices like disk. There are twokinds of files in a system. They are,

1. Text files (ASCII)
2. Binary files

- Text files contain ASCII codes of digits, alphabetic and symbols.
- Binary file contains collection of bytes (0‟s and 1‟s). Binary files are compiled version of text files.

Basic File Operations

There are 4 basic operations that can be performed on any files in C programming language. They are,

1. Opening/Creating a file
2. Closing a file
3. Reading a file
4. Writing in a file

Let us see the syntax for each of the above operations in a table:

File operation	Declaration & Description
fopen() – To open a file	Declaration: FILE ***fopen** (const char *filename, const char *mode) fopen() function is used to open a file to perform operations such as reading, writing etc. In a C program, we declare a file pointer and use fopen() as below. fopen() function creates a new file if the mentioned file name does not exist. FILE *fp; fp=**fopen** (“filename”, ”‘mode”); Where, fp – file pointer to the data type “FILE”. filename – the actual file name with full path of the file. mode – refers to the operation that will be performed on the file. Example: r, w, a, r+, w+ and a+. Please refer below the description for these mode of operations.

fclose() – To close a file	Declaration: int **fclose**(FILE *fp); fclose() function closes the file that is being pointed by file pointer fp. In a C program, we close a file as below. **fclose** (fp);
fgets() – To read a file	Declaration: char ***fgets**(char *string, int n, FILE *fp) fgets function is used to read a file line by line. In a C program, we use fgets function as below. **fgets** (buffer, size, fp); where, buffer – buffer to put the data in. size – size of the buffer fp – file pointer
fprintf() – To write into a file	Declaration: int **fprintf**(FILE *fp, const char *format, ...);fprintf() function writes string into a file pointed by fp. In a C program, we write string into a file as below. fprintf (fp, "some data"); or fprintf (fp, "text %d", variable_name);

Module Description:

1. login():-The login() Function enables the user to access their preexisting data from the file. The user is asked to enter their id and password. We search for an account with the id. If the respective password matches the user entered pass we grant the user access to the details.

2. details():-The details() Function gets the details from the user and stores then in the structure, data. It reads the candidate name, gender, date of birth, age, Aadhaar number, Email id, Mobile number and number of dose already taken and stores them in respective structure members. On getting all values from the user It calls the generate id Function To create an account for the user. It also gets the details about the venue. Finally, it uses the add data Function to append all the obtained information to the file.

3. The venue of module allows the user to select the most convenient from a list of options to get the vaccination. The module allows the user to select the date and time before providing

the user a list of Hospitals. The user is asked to enter their choice. The information is stored in a global variable.

4. printdetails():-The print details module as it's name implies prints the details of the user. It displays the name, Gender, age, Date of Birth, mobile number, Email Id, Number of doses taken and the Venue for vaccination.

5. generateid():-The generateid Function, as it's name implies generates Register number for a new user. It checks the register number of the previous Candidate and generates a new Register number by incrementing the Register number of the previous Candidate by 1. It also allows the user to create a new password for their account and store the password along with the generatedid in the accountid structure.

6. adddata():-The adddata() adds the new candidate's information to the preexisting file which contains data of other Candidates. It retrieves the name, age, gender, mobile number, Email Id, Aadhaar Id, date of birth, number of doses taken from the structure and appends it to the end of the file.

7. getdetails():-The getdetails() Function return the corresponding data present in the file when given an Register number. It searches the file for the said Register number and retrieves the values of name, age, gender, mobile number, Email Id, Aadhaar Id, date of birth, number of doses taken and venue from the file and saves them in the structure named data.

Implementation:

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <string.h>

struct data {
   char regid[10];
   char name[20];
   char age[10];
   char g[10];
   char dob[20];
   char aid[20];
   char mob[20];
   char eid[100];
   char dose[2];
} dt;
struct accountid {
   char id[10];
   char pass[16];
} idt;

int n,c=0;
char state[20], dis[20], hos[40], date[12], hour[6];

void login();
void details();
void venue();
void printdetails();
void generateid();
void addata();
void getdetails(char* id);

int main()
{  int ch;
   printf("***Vaccine Registration System\n***");
   printf("Please choose one of the following actions:\n1.Register for Vaccination\n
   2.Check Your Status\n3.Exit\n\nPlease Enter Your Choice: ");
   scanf("%d",&ch);
   switch(ch)
   {
   case 1:
   details();
   printdetails();
   break;
```

```
   case 2:
    login();
    printdetails();
    break;
    case 3:
    break;
    default:
    printf("No such choice exists!");
   }
   return 0;
}
 void login(){
 char id[10],pass[16],*token,row[100];
 FILE *fp;
 printf("Enter your registration id: ");
 fgets(id, sizeof(id), stdin);
 printf("Enter your Password:");
 fgets(pass, sizeof(pass), stdin);
 fp=fopen("id.csv","r");
 while(!feof(fp)){
 fgets(row,100,fp);
 token=strtok(row,",");
 if(strcmp(token,id)==0){
 strcpy(idt.id,token);
 token=strtok(NULL,",");
 strcpy(idt.pass,token);}
 }
 while(1){
 if(strcmp(idt.pass,pass)==10){
 printf("Successfully logged in...\nFetching your details...");
 getdetails(id);
 break;
 }
 else{
 printf("Wrong Password! Try Again.");
 scanf("%s",pass);}
 }
 getdetails(id);
 }
```

```
void getdetails(char *id){
char row[100],*token;
FILE *fp;
fp=fopen("NMYFILE.csv","r");
while(!feof(fp)){
fgets(row,100,fp);
token=strtok(row,",");
if(strcmp(token,id)==0)
break;
}
strcpy(dt.regid,token);
while(token){
token=strtok(NULL,",");
strcpy(dt.name,token);
token=strtok(NULL,",");
strcpy(dt.g,token);
token=strtok(NULL,",");
strcpy(dt.age,token);
token=strtok(NULL,",");
strcpy(dt.aid,token);
token=strtok(NULL,",");
strcpy(dt.mob,token);
token=strtok(NULL,",");
strcpy(dt.eid,token);
token=strtok(NULL,",");
strncpy(dt.dose, token, sizeof(dt.dose));}
}

void addata(){
   FILE *fpt = fopen("NMYFILE.csv", "a");
   if (fpt == NULL) {
      printf("Failed to open the file for writing.\n");
      return;}
   dt.name[strcspn(dt.name, "\n")] = '\0';
   dt.g[strcspn(dt.g, "\n")] = '\0';
   dt.dob[strcspn(dt.dob, "\n")] = '\0';
   dt.age[strcspn(dt.age, "\n")] = '\0';
   dt.aid[strcspn(dt.aid, "\n")] = '\0';
   dt.mob[strcspn(dt.mob, "\n")] = '\0';
   dt.eid[strcspn(dt.eid, "\n")] = '\0';
   dt.dose[strcspn(dt.dose, "\n")] = '\0';
   fprintf(fpt, "%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s\n", dt.regid, dt.name, dt.g,
   dt.dob, dt.age, dt.aid, dt.mob, dt.eid, dt.dose, hos, date, hour);
   fclose(fpt);
}
```

```
void details(){
   printf("\t\tEnter The Candidate Name: ");
   fflush(stdin);
   fgets(dt.name,20,stdin);
   printf("\t\tEnter The Candidate Gender: ");
   fflush(stdin);
   fgets(dt.g,10,stdin);
   printf("\t\tEnter the Candidate date of birth: ");
   fflush(stdin);
   fgets(dt.dob,20,stdin);
   printf("\t\tEnter the Candidate Age: ");
   fflush(stdin);
   fgets(dt.age,10,stdin);
   printf("\t\tEnter The Candidate Aadhaar Number: ");
   fflush(stdin);
   fgets(dt.aid,20,stdin);
   printf("\t\tEnter The Candidate Email Id: ");
   fflush(stdin);
   fgets(dt.eid,20,stdin);
   printf("\t\tEnter The Candidate Mobile Number: ");
   fflush(stdin);
   fgets(dt.mob,100,stdin);
   printf("\t\tEnter the number of dose already taken: ");
   fflush(stdin);
   fgets(dt.dose,2,stdin);
   generateid();
   venue();
   addata();
}

void generateid(){
   FILE *fp = fopen("id.csv", "a+");
   if (fp == NULL) {
      printf("Failed to open the file for writing.\n");
      return;
   }
   char row[100], *token, t[10];
   int i;
   while (!feof(fp))
      fgets(row, sizeof(row), fp);
   token = strtok(row, ",");
   strcpy(t, token);
   while (token)
      token = strtok(NULL, ",");
```

```
    for (i = 0; t[i] != '\0'; i++);
    if ((int)(t[i - 1]) == 57)
        t[i - 1] = 48;
    else
        t[i - 1]++;
    strcpy(idt.id, t);
    printf("\t\tCreate your password: ");
    scanf("%s", idt.pass);
    fprintf(fp, "%s,%s\n", idt.id, idt.pass);
    fclose(fp);
    strcpy(dt.regid, idt.id);
}

void venue(){
    int i;
    system("cls");
    printf("Please select Your Venue:\n");
    printf("\n\nChoose one of the following hospitals for Vaccination:\n");
    printf("\t\t1. Rio Hospital\n");
    printf("\t\t2. Apollo Hospital\n");
    printf("\t\t3. Lakeshore Hospital\n");
    printf("\t\tEnter Choice: ");
    scanf("%d", &i);
    if (i == 1)
        strcpy(hos, "Rio Hospital");
    else if (i == 2)
        strcpy(hos, "Apollo Hospital");
    else if (i == 3)
        strcpy(hos, "Lakeshore Hospital");
    else
        printf("Enter Correct Choice...");
    fflush(stdin);
    printf("\t\tEnter Date (DD-MM-YY): ");
    gets(date);
    printf("\t\tEnter Time (24 Hours): ");
    gets(hour);
}
```

```
void printdetails()
{
   system("cls");
   printf("Your Details:\n");
   printf("Register No.: %s\n",dt.regid);
   printf("Name: %s\n",dt.name);
   printf("Gender: %s\n",dt.g);
   printf("Age: %s\n",dt.age);
   printf("Date of Birth: %s\n",dt.dob);
   printf("Aadhaar Id: %s\n",dt.aid);
   printf("Mobile No.: %s\n",dt.mob);
   printf("Email Id: %s\n",dt.eid);
   printf("Dose: %s\n",dt.dose);
   printf("Hospital: %s\n",hos);
   printf("Date: %s \t Time: %s",date,hour);
}
```

Output:

```
***Vaccine Registration System
***Please choose one of the following actions:
1.Register for Vaccination
2.Check Your Status
3.Exit

Please Enter Your Choice: 1

                Enter The Candidate Name: Joey
                Enter The Candidate Gender: Male
                Enter the Candidate date of birth: 23-05-2001
                Enter the Candidate Age: 22
                Enter The Candidate Aadhaar Number: 445433423456
                Enter The Candidate Email Id: joeystan@gmail.com
                Enter The Candidate Mobile Number: 6546348905
                Enter the number of dose already taken: 1
                Create your password: kjksfdnnkfS
```

```
Please select Your Venue:
Choose one of the following hospitals for Vaccination:
                1. Rio Hospital
                2. Apollo Hospital
                3. Lakeshore Hospital
                Enter Choice: 1
                Enter Date (DD-MM-YY): 28-06-2023
                Enter Time (24 Hours): 14:30S
```

```
Your Details:
Register No.: 117
Name: Joey
Gender: Male
Age: 22
Date of Birth: 23-05-2001
Aadhaar Id: 445433423456
Mobile No.: 6546348905
Email Id: joeystan@gmail.com
Dose: 1
Hospital: Rio Hospital
Date: 28-06-2023          Time: 14:30_
```

	A	B	C	D	E	F	G
1	ID	pass					
2	111	qwer					
3	112	asdf					
4	113	zxcvkipl					
5	114	ibjkgjkk					
6	115	sgsdggjk					
7	116	asdfdgjk					
8	117	kjksfdnnkfS					

	A	B	C	D	E	F	G	H	I	J	K	L
1	Regid	Name	Gender	DOB	Age	Aadhaar ID	Mobile No.	Email ID	Dose	Hospital	Date	Time
2	111	Chris	Male	17.07.1978	44	697387545447	7091567503	chrishowa	1	Rio Hospit	16-06-202	13:30
3	112	Britney	Female	04.07.2000	22	381255449798	1865881312	britneysp	1	Apollo Ho	17-06-202	12:20
4	113	Kimberly	Female	04.08.1960	62	924125453375	3914123157	kim1960@	2	Rio Hospit	17-06-202	19:30
5	114	Louis	Male	05.05.2004	19	526545238147	7585628825	louishami	1	Apollo Ho	23-06-202	12:20
6	115	Harry	Male	09.02.2000	23	431389932242	7647242391	harrystyle	1	Lakeshore	22-06-202	19:30
7	116	Karthik	Male	16.06.1998	25	160832204062	8362433672	karthik19	1	Rio Hospit	27-06-202	12:20
8	117	Joey	Male	23-05-200	22	445433423456	6546348905	joeystan@	1	Rio Hospit	28-06-202	14:30

Result:

In this mini project, we have developed vaccine registration system in C and the output was verified successfully.

Stepwise Execution:

- At the top of the program we include the necessary header files for the program. We then declare the structures data and account id. Then we declare the global variables and finally we declare the functions.
- Inside the main Function we first ask the user the select their next course of action. Whether they want to register for Vaccine or check the status or exit the program.
- If the user chooses to register for Vaccine we call the details and print details Function.
- If the user chooses to check the status we call the login and printdetails Function.
- If the Choice of the user doesn't exist we display a message that says no such Choice exists.

- If the user Enters 1, Then the details Function is called. Inside the details function we get the candidate name, gender, date of birth, age, Aadhaar number, Email id, Mobile number and number of dose already taken from the user using fgets function. After getting all values we call generateid function.
- The generateid function opens the id.csv file which contains the register number and password of the user It reads each row in the file till it reaches EOF using a while loop. On reaching EOF it exits the loop. The value of row would be the last row of the file. We then use the strtok Function To tokenize the string row. We separate it into token using the delimiter '",". The first token of row would be the register number and is copied to the character array t. The final element of the array is found using for loop. If the last element of array is 9 we change it into 0 and the adjacent element is increased by 1. Else we increase the last element by 1. Then we ask the user to create password. We save the id and password in idt.id and idt.pass. After closing the file pointer the control finally return back to statement after the function call.
- The venue() Function is now called. In this we let the user choose the hospital he/she is convenient with from the mentioned hospitals. We also let them choose the date and time they are comfortable with.
- On returning back from the venue() Function the addata Function is now called. The addata Function opens the file in append mode. It add the data stored in the data structure using the structure variable dt to access the member of the structure to the file using the fprintf function. After it completes it's function the file pointer is closed. Now the printdetails function is called. This function prints all the data collected so far about the user in the file. It prints the register number, name, gender, age, date of birth, Aadhaar Id, Mobile number, Email Id, Dose, Hospital and Date.
- If the user enters 2 and chooses to check status, the login Function is called. It asks the user to enter the register number and password. It searches the id. csv which stores the relevant content for the registration number and save it in idt.id the member of accountid structure. The corresponding password is stored in idt.pass and is compared with user input. If the passwords match the user is allowed to get their details else it asks for the user to enter the password again.
- The getdetails Function accepts the registration number as an argument and searches for the number in MYFILE.csv. On finding the corresponding row it saves the data in the structure. The control return to the main function and the following printdetails Function is executed.It now prints the data read from the file. Finally, if the user enters 3 and chooses to exit from the program we break statement to exit the switch case. If the users Enters a choice which does not match any of the given choices we print a message that says that the Choice doesn't exist and exit the program.

Conclusion:

The development of vaccine registration system has been successful. We have created an easy and understandable program which enables the user to use the registration system more efficiently. We have used a lot of features and functionalities to meet the needs and expectation.