M.H SABOO SIDDIK POLYTECHNIC

DATA STRUCTURE MICRO-PROJECT.

-UNDER THE GUIDANCE OF

SHAFAQUE MA'AM.

TOPIC: COVID-19 RELATED INFORMATION.

PRESENTED BY :

SR.NO	ROLL NO	NAME
1.	19401	Ansari Anam.
2.	19416	Loladia Ayesha.
3.	19420	Rakhangi Arisha.
4.	19428	Shaikh Nusra.

MAHARASHTRA STATE
BOARD OF TECHNICAL EDUCATION
CERTIFICATE

This is to certify that

Mr/Ms: _____

Roll No: _____ Enrollment No: _____

of Third Semester of Diploma in **Computer Engineering** of Institute

M. H. Saboo Siddik Polytechnic (Code **: 0002**) has completed their term work satisfactorily in course **Data Structure Using C(22317)** for the academic year 2020 to 2021 as prescribed in the curriculum.

Place : Mumbai Date : .....

Exam. Seat No : .....

Subject Teacher. Head of the Department. Principal

Seal of Institute

MAHARASHTRA STATE
BOARD OF TECHNICAL EDUCATION
CERTIFICATE

This is to certify that

Mr/Ms: _____

Roll No: _____ Enrollment No: _____

of Third Semester of Diploma in **Computer Engineering** of Institute

M. H. Saboo Siddik Polytechnic (Code **: 0002**) has completed their term work satisfactorily in course **Data Structure Using C(22317)** for the academic year 2020 to 2021 as prescribed in the curriculum.

Place : Mumbai Date : .....

Exam. Seat No : .....

Subject Teacher. Head of the Department. Principal

Seal of Institute

MAHARASHTRA STATE
BOARD OF TECHNICAL EDUCATION
CERTIFICATE

This is to certify that

Mr/Ms: _____

Roll No: _____ Enrollment No: _____

of Third Semester of Diploma in **Computer Engineering** of Institute

M. H. Saboo Siddik Polytechnic (Code **: 0002**) has completed their term work satisfactorily in course **Data Structure Using C(22317)** for the academic year 2020 to 2021 as prescribed in the curriculum.

Place : Mumbai Date : .....

Exam. Seat No : .....

Subject Teacher. Head of the Department. Principal

Seal of Institute

MAHARASHTRA STATE
BOARD OF TECHNICAL EDUCATION
CERTIFICATE

This is to certify that

Mr/Ms: _____

Roll No: _____ Enrollment No: _____

of Third Semester of Diploma in **Computer Engineering** of Institute

M. H. Saboo Siddik Polytechnic (Code **: 0002**) has completed their term work satisfactorily in course **Data Structure Using C(22317)** for the academic year 2020 to 2021 as prescribed in the curriculum.

Place : Mumbai Date : .....

Exam. Seat No : .....

Subject Teacher. Head of the Department. Principal

Seal of Institute

(NAME AND SIGNATURE OF FACULTY)

ANNEXURE

Evaluation sheet for the micro-project.

Academic year: 2020-21. **Name of faculty: Shafaque Ma'am.**

Course: DSU. **Course code: 22317.** **Semester: III.**

Title of the project:

CO's addressed by the micro-project:

A. _____

B. _____

Major learning outcomes achieved by the students by doing the project :

(a) Practical Outcomes: _____

_____

_____

(b) Unit outcomes in cognitive domain: _____

_____

_____

(c) Outcomes in effective domain: _____

_____

_____

Comments/ suggestions about teamwork/ leadership/ inter-personal communication (if any):

Roll No:	Student Name	Marks out of 6 for performance in group activity	Marks out of 4 for performance in oral or presentation	Total out of 10
19401	Ansari Anam			
19416	Loladia Ayesha			
19420	Rakhangi Arisha			
19428	Shaikh Nusra			

In Our Program We Have Used The Following Topics :

1)QUEUE : Queues are linear data structures in which we add elements to one end and remove them from the other end. The first item to be inserted (enqueue) is the first to be deleted (dequeue). A queue is therefore called a **F**irst **I**n **F**irst **O**ut (FIFO).

Queue On operations:

1)Enqueue: Insert an element at the rear of the queue.

2)Dequeue: Delete an element at the front of the queue.

➢ Enqueue Operation:

Queues maintain two data pointers, front and rear. Therefore, its operations are comparatively difficult to implement than that of stacks.

The following steps should be taken to enqueue (insert) data into a queue −

Step 1 − Check if the queue is full.

Step 2 − If the queue is full, produce overflow error and exit.

Step 3 − If the queue is not full, increment rear pointer to point the next empty space.

Step 4 − Add data element to the queue location, where the rear is pointing.

Step 5 − return success.

➢ Dequeue Operation:

Accessing data from the queue is a process of two tasks − access the data where front is pointing and remove the data after access. The following steps are taken to perform dequeue operation −

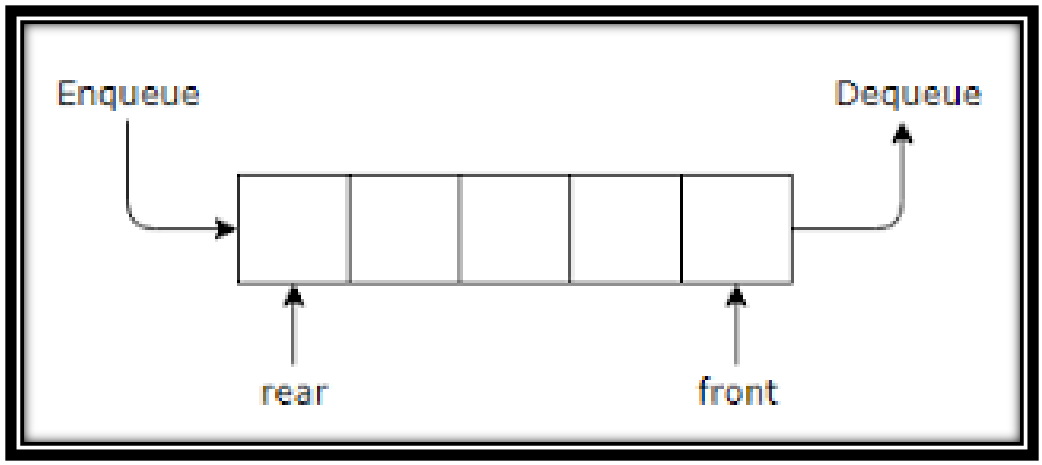
Step 1 − Check if the queue is empty.

Step 2 − If the queue is empty, produce underflow error and exit.

Step 3 − If the queue is not empty, access the data where front is pointing.

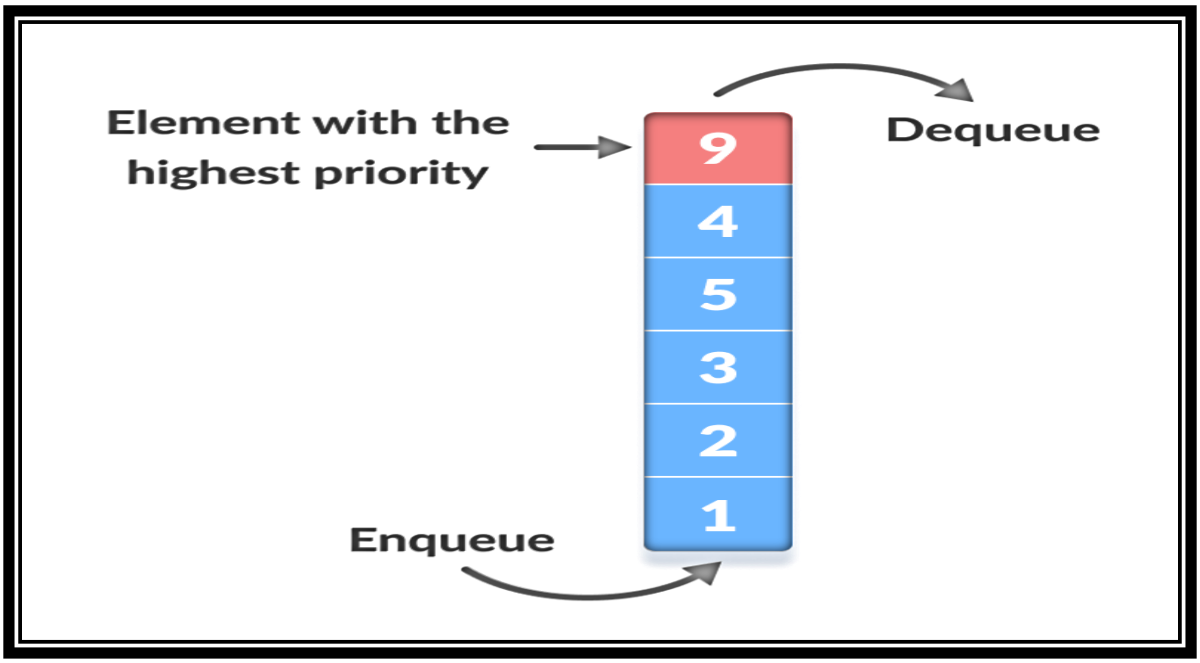
Step 4 − Increment front pointer to point to the next available data element.

Step 5 − Return success.



2) PRIORITY QUEUE: A priority queue is a special type of queue in which each element is associated with a priority and is served according to its priority. If elements with the same priority occur, they are served according to their order in the queue. Generally, the value of the element itself is considered for assigning the priority.

For example, the element with the highest value is considered as the highest priority element. However, in other cases, we can assume the element with the lowest value as the highest priority element. In other cases, we can set priorities according to our needs.



3)Linked List : A linked list is a linear data structure, in which the elements are not stored at contiguous memory locations. A linked list consists of nodes where each node contains a data field and a reference (link) to the next node in the list. The elements in a linked list are linked using pointers as shown in the below image:

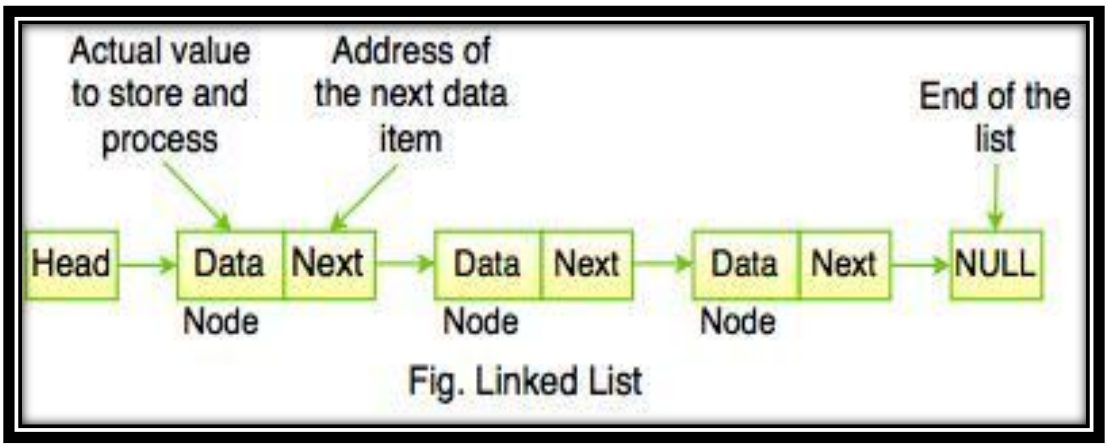


Fig. Linked List

- ## Some Terminologies In Linked List:

1) **NODE**: A Node in a linked list holds the data value and the pointer which points to the location of the next node in the linked list.
2) **NEXT:** It contains a pointer to the next element or node.
3) **NULL POINTER:** The next field of the last node contain NULL rather than address. It indicates end of the list.
4) **HEAD:** It contains the address of the first node.

Aim: Queue Using Link List & Priority Queue To Sort The Data.

Code:

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

void worldcases();   //queue using link list
void displayworld(); //
void deleteworld();
void usacases();     //
void displayusa();   //
void deleteusa();
void indiacases();   //
void displayindia(); //
void deleteindia();
void francecases();  //
void displayfrance();//
void deletefrance();
void brazilcases();  //
void displaybrazil();//
void deletebrazil();
void netherlandscases();//
void displaynetherlands();//
void deletenetherlands();
void japancases();    //
void displayjapan();  //
void deletejapan();
void skoreacases();   //
void displayskorea(); //
void deleteskorea();
void vacc_names();    // priority queue using linked list
void displayvacc_name();
void lab_names();     //
void displaylab_name();  //
void man_names();        //
void displayman_name(); //
void testing_sub();
void display_sub();

struct world *front=NULL,*rear=NULL; //world
struct world *front1=NULL,*rear1=NULL; //usa
struct world *front2=NULL,*rear2=NULL; //india
struct world *front3=NULL,*rear3=NULL; //france
struct world *front4=NULL,*rear4=NULL; //brazil
struct world *front5=NULL,*rear5=NULL; //netherlands
struct world *front6=NULL,*rear6=NULL; //japan
struct world *front7=NULL,*rear7=NULL; //south korea
struct vacc *frontq1=NULL,*rearq1=NULL; //vaccine names
```

```
struct vacc *frontq2=NULL,*rearq2=NULL;
//laboratory names

struct vacc *frontq3=NULL,*rearq3=NULL;
//manufacturer names

struct vacc *frontq4=NULL,*rearq4=NULL;
//testing subjects

int i1=1;//world

int i2=1;//usa

int i3=1;//india

int i4=1;//france

int i5=1;//brazil

int i6=1;//netherlands

int i7=1;//japan

int i8=1;//south korea



typedef struct world

{

 unsigned long int data;

 struct world *link;

}world;

void worldcases()

{

 world *ptr;

 unsigned long int
a[6]={62004949,42798107,17757534,105249,1
449308,7828430350};

 int i;

 front=(world*)malloc(sizeof(world));

 front->data=a[0];

 front->link=NULL;

 rear=front;

 if(i1==1)

 {

 for(i=1;i<6;i++)

 {

  ptr=(world*)malloc(sizeof(world));

  ptr->data=a[i];

  ptr->link=NULL;

  rear->link=ptr;

  rear=ptr;

 }

 }

 else if(i1==2)

 {

  for(i=2;i<6;i++)

 {

  ptr=(world*)malloc(sizeof(world));

  ptr->data=a[i];

  ptr->link=NULL;

  rear->link=ptr;

  rear=ptr;

 }

 }

 else if(i1==3)

 {

  for(i=3;i<6;i++)

 {

  ptr=(world*)malloc(sizeof(world));

  ptr->data=a[i];

  ptr->link=NULL;

  rear->link=ptr;
```

```
 rear=ptr;
}
}
else if(i1==4)
{
 for(i=4;i<6;i++)
{
 ptr=(world*)malloc(sizeof(world));
 ptr->data=a[i];
 ptr->link=NULL;
 rear->link=ptr;
 rear=ptr;
}
}
else if(i1==5)
{
 for(i=5;i<6;i++)
{
 ptr=(world*)malloc(sizeof(world));
 ptr->data=a[i];
 ptr->link=NULL;
 rear->link=ptr;
 rear=ptr;
}
}
 displayworld();
}
```

Similarly With respect to above code , create similar functions named as usacases(), indiacases() , francecases() , brazilcases() , netherlandscases(), japancases() , skoreacases() respectively with different records in every function .

```
void displayworld()
{
 world *temp;
 temp=front;
 if(i1==1)
 {
  printf("****************** World Wide COVID Infections **************\n");
  printf("\n");
  printf("cases  |  recovered |  active | serious | deaths |  population | \n");
  printf("-------------------------------------------------------------\n");
  while(temp!=NULL)
  {
   printf("%lu | ",temp->data);
   temp=temp->link;
  }
 }
 else if (i1==2)
 {
  printf("****************** World Wide COVID Infections **************\n");
  printf("\n");
  printf(" recovered |  active | serious | deaths | population | \n");
  printf("-------------------------------------------------------------\n");
  while(temp!=NULL)
```

```
 {
  printf("%lu | ",temp->data);
  temp=temp->link;
 }
}
 else if (i1==3)
 {
  printf("****************** World Wide COVID Infections **************\n");
  printf("\n");
  printf(" active | serious | deaths |  population | \n");
  printf("-----------------------------------------\n");
  while(temp!=NULL)
  {
   printf("%lu | ",temp->data);
   temp=temp->link;
  }
 }
 else if(i1==4)
 {
  printf("****************** World Wide COVID Infections **************\n");
  printf("\n");
  printf(" serious | deaths |  population | \n");
  printf("-----------------------------------\n");
  while(temp!=NULL)
  {
   printf("%lu | ",temp->data);
   temp=temp->link;
  }
 }
 else if(i1==5)
 {
  printf("****************** World Wide COVID Infections **************\n");
  printf("\n");
  printf(" deaths |  population | \n");
  printf("-------------------------\n");
  while(temp!=NULL)
  {
   printf("%lu | ",temp->data);
   temp=temp->link;
  }
 }
 else if(i1==6)
 {
  printf("****************** World Wide COVID Infections **************\n");
  printf("\n");
  printf(" population | \n");
  printf("---------------\n");
  while(temp!=NULL)
  {
   printf("%lu | ",temp->data);
   temp=temp->link;
  }
 }
 else
 printf("queue is empty\n");
}
```

With respect to above code, create similar functions named as displayusa() , displayindia() , displayfrance() , displaybrazil() , displaynetherlands() , displayjapan() , displayskorea() .

```
void deleteworld()
{
 struct world *temp;
 temp=front;
 front=front->link;
 printf("\n\n%d field was deleted from the queue\n\n",i1);
 i1++;
 free(temp);
 displayworld();
}
```

With respect to above code, create similar functions named as deleteusa() , deleteindia() , deletefrance() , deletebrazil() , deletenetherlands() , deletejapan() , deleteskorea() .

```
typedef struct vacc
{
 char name[30];
 int p;
 struct node *link;
}vacc;
void displayvacc_name()
{
 struct vacc *temp;
 temp=frontq1;
 printf("LIST OF COVID VACCINES:.....\n");
 printf("\n");
 printf("vaccine name                priority\n");
 printf("_____\n");
 while (temp!=NULL)
 {
  printf("%s vaccine                %d\n",temp->name,temp->p);
  temp=temp->link;
 }
}
void vacc_names()
{
 vacc *ptr;
 clrscr();
 frontq1=(vacc*)malloc(sizeof(vacc));
 strcpy(frontq1->name,"Pneumocal");
 frontq1->p=1;
 frontq1->link=NULL;
 rearq1=frontq1;
 ptr=(vacc*)malloc(sizeof(vacc));
 strcpy(ptr->name,"hemophils");
 ptr->p=2;
 ptr->link=NULL;
 rearq1->link=ptr;
 rearq1=ptr;
 ptr=(vacc*)malloc(sizeof(vacc));
 strcpy(ptr->name,"pneumovac");
 ptr->p=3;
```

```
ptr->link=NULL;
rearq1->link=ptr;
rearq1=ptr;
ptr=(vacc*)malloc(sizeof(vacc));
strcpy(ptr->name,"pertusis ");
ptr->p=4;
ptr->link=NULL;
rearq1->link=ptr;
rearq1=ptr;
ptr=(vacc*)malloc(sizeof(vacc));
strcpy(ptr->name,"anthrax  ");
ptr->p=5;
ptr->link=NULL;
rearq1->link=ptr;
rearq1=ptr;
ptr=(vacc*)malloc(sizeof(vacc));
strcpy(ptr->name,"anciphalt");
ptr->p=6;
ptr->link=NULL;
rearq1->link=ptr;
rearq1=ptr;
ptr=(vacc*)malloc(sizeof(vacc));
strcpy(ptr->name,"flu shot ");
ptr->p=7;
ptr->link=NULL;
rearq1->link=ptr;
rearq1=ptr;
displayvacc_name();
}
```

Similarly With respect to above code, create similar functions named as lab_names() , man_names(), testing_sub() and call displaylab_name(), displayman_name() , display_sub() respectively with different records in every function.

```
void main()
{
 int ch;//main menu
 for(;;)
 {
  clrscr();
  printf("****************************** Data Structure Project *************************\n");
  printf("****************************** COVID-19 information *************************\n");
  printf("******************** MAIN MENU ******************\n");
  printf("ENTER YOUR CHOICE\n");
  printf("1. Display Menu\n");
  printf("2. Delete menu\n");
  printf("3. EXIT \n");
  scanf("%d",&ch);
  switch(ch)
  {
   case 1://display menu
   clrscr();
   printf("******************** DISPLAY MENU ******************\n");
   printf("ENTER YOUR CHOICE\n");
```

```
  printf("1. COVID Infection Cases information
\n");

  printf("2. COVID vaccination information\n");

  printf("3. Return to Main Menu\n");

  printf("4. EXIT\n");

  scanf("%d",&ch);

  switch(ch)

  {

   case 1:

   clrscr();

   printf("**************** COVID INFECTION
CASES INFO MENU ****************\n");

   printf("ENTER YOUR CHOICE\n");

   printf("1. Display Total World Wide cases\n");

   printf("2. Display info of any Specific
country\n");

   printf("3. Return to Main Menu\n");

   printf("4. EXIT\n");

   scanf("%d",&ch);

   switch(ch)

   {

    case 1:

    clrscr();

    worldcases();

    getch();

    break;

    case 2:

    clrscr();

    printf("*************** Country Wise COVID
Infections ***************\n");

    printf("ENTER YOUR CHOICE\n");

    printf("1. USA\n");

    printf("2. India\n");

    printf("3. France\n");

    printf("4. Brazil\n");

    printf("5. Netherlands\n");

    printf("6. Japan\n");

    printf("7. South Korea\n");

    printf("8. Return to main menu\n");

    printf("9. EXIT\n");

    scanf("%d",&ch);

    switch(ch)

    {

     case 1://usa

     clrscr();

     usacases();

     getch();

     break;

     case 2://india

     clrscr();

     indiacases();

     getch();

     break;

     case 3://france

     clrscr();

     francecases();

     getch();

     break;

     case 4://brazil

     clrscr();

     brazilcases();
```

```
 getch();
 break;
 case 5://netherlands
 clrscr();
 netherlandscases();
 getch();
 break;
 case 6://japan
 clrscr();
 japancases();
 getch();
 break;
 case 7://south korea
 clrscr();
 skoreacases();
 getch();
 break;
 case 8:
 break;
 case 9:
 exit(0);
 default:
 printf("Invalid Choice Try Again\n");
 break;
 }
 break;
}
break;
case 2://vaccine info
clrscr();
   printf("******************** COVID Vaccination Info Menu ********************\n");
   printf("ENTER YOUR CHOICE\n");
   printf("1. Display list of vaccine names\n");
   printf("2. Display list of laboratory names involved in covid vaccine creation\n");
   printf("3. Display list of manufacturer names involved in vaccine creation\n");
   printf("4. Display list of testing subjects used in vaccine creation\n");
   printf("5. Return to main menu\n");
   printf("6. EXIT\n");
   scanf("%d",&ch);
   switch(ch)
   {
    case 1: //vaccine names
    clrscr();
    vacc_names();
    getch();
    break;
    case 2: //lab names
    clrscr();
    lab_names();
    getch();
    break;
    case 3://manufacturer names
    clrscr();
    man_names();
    getch();
    break;
    case 4://testing subjects
```

```
clrscr();
testing_sub();
getch();
break;
case 5:
break;
case 6:
exit(0);
break;
default:
printf("Invalid Choice Try Again\n");
break;
}
break;
case 3://return to main menu
break;
case 4:
exit(0);
default:
printf("Invalid choice Try Again\n");
break;
}
break;
case 2:// edit menu
clrscr();
printf("******************** Delete Menu ********************\n");
printf("Enter your choice\n");
printf("1. Delete covid infections record\n");
printf("2. Return to main menu\n");
printf("3. EXIT\n");
scanf("%d",&ch);
switch(ch)
{
case 1: //covid infections menu
clrscr();
printf("******************** COVID Infection Records ********************\n");
printf("Choose from the following records to edit\n");
printf("1. world covid record\n");
printf("2. USA covid record\n");
printf("3. India covid record\n");
printf("4. France covid record\n");
printf("5. Brazil covid record\n");
printf("6. Netherlands covid record\n");
printf("7. Japan covid record\n");
printf("8. South Korea covid record\n");
printf("9. Return to main menu\n");
printf("10. EXIT\n");
scanf("%d",&ch);
switch(ch)
{
case 1: //world
clrscr();
worldcases();
deleteworld();
getch();
break;
case 2: //usa
clrscr();
```

```
usacases();
deleteusa();
getch();
break;
case 3: //india
clrscr();
indiacases();
deleteindia();
getch();
break;
case 4: //france
clrscr();
francecases();
deletefrance();
getch();
break;
case 5: //brazil
clrscr();
brazilcases();
deletebrazil();
getch();
break;
case 6: //netherlands
clrscr();
netherlandscases();
deletenetherlands();
getch();
break;
case 7: //japan
clrscr();
japancases();
deletejapan();
getch();
break;
case 8: //south korea
clrscr();
skoreacases();
deleteskorea();
getch();
break;
case 9:
break;
case 10:
exit(0);
default:
printf("Invalid Choice Try Again\n");
break;
}
case 2:
break;
case 3:
exit(0);
default:
printf("invalid choice try again\n");
break;
}
getch();
break;
case 3:
exit(0);
```

```
 break;
 default:
 printf("Invalid choice Try again\n");
 break;
 }
}
```

```
 getch();
}
```

OUTPUT:

```
********************************** Data Structure Project **********************

********************************** COVID-19 information ************************

********************** MAIN MENU *********************
ENTER YOUR CHOICE
1. Display Menu
2. Delete menu
3. EXIT
```

```
********************** DISPLAY MENU **********************
ENTER YOUR CHOICE
1. COVID Infection Cases information
2. COVID vaccination information
3. Return to Main Menu
4. EXIT
_
```

```
****************** COVID INFECTION CASES INFO MENU ******************
ENTER YOUR CHOICE
1. Display Total World Wide cases
2. Display info of any Specific country
3. Return to Main Menu
4. EXIT
```

```
****************** World Wide COVID Infections ***************

cases  |  recovered |  active | serious | deaths |  population |
-------------------------------------------------------------------
62004949 | 42798107 | 17757534 | 105249 | 1449308 | 3533463054 |
```

```
********** Country Wise COVID Infections **********
ENTER YOUR CHOICE
1. USA
2. India
3. France
4. Brazil
5. Netherlands
6. Japan
7. South Korea
8. Return to main menu
9. EXIT
```

```
********** USA COVID Infections **********

cases  |  recovered |  active | serious | deaths |  population |
-------------------------------------------------------------------
10182818  |  3497817  |  18480  |  6441744  |  243257  |  331690410  |  _
```

```
********** India COVID Infections **********

cases  |  recovered |  active | serious | deaths |  population |
-------------------------------------------------------------------
850774  |  512624  |  8944  |  7868968  |  126162  |  1384789754  |
```

```
********** France COVID Infections **********

cases  |  recovered |  active | serious | deaths |  population |
-------------------------------------------------------------------
1748705  |  1580598  |  4421  |  127938  |  40169  |  65324901  |
```

```
********** Brazil COVID Infections **********

cases  |  recovered |  active | serious | deaths |  population |
-------------------------------------------------------------------
5653561  |  426931  |  8318  |  5064344  |  162286  |  213094038  |
```

```
****************** Netherlands COVID Infections ***************

cases  |  recovered |  active | serious | deaths |  population |
-------------------------------------------------------------------
404401   |  2000   |  630   |  300205   |  7690   |  17148361   |
```

```
****************** Japan COVID Infections ***************

cases  |  recovered |  active | serious | deaths |  population |
-------------------------------------------------------------------
105941   |   7644   |   194   |   96461   |   1809   |   126337911   |
```

```
****************** South Korea COVID Infections ***************

cases  |  recovered |  active | serious | deaths |  population |
-------------------------------------------------------------------
27427   |   1981   |   58   |   24968   |   478   |   51284907   |   _
```

```
******************** COVID Vaccination Info Menu *********************
ENTER YOUR CHOICE
1. Display list of vaccine names
2. Display list of laboratory names involved in covid vaccine creation
3. Display list of manufacturer names involved in vaccine creation
4. Display list of testing subjects used in vaccine creation
5. Return to main menu
6. EXIT
```

```
LIST OF COVID VACCINES:........

vaccine name                    priority
_____________________________________
Pneumocal vaccine                  1
hemophils vaccine                  2
pneumovac vaccine                  3
pertusis  vaccine                  4
anthrax   vaccine                  5
anciphalt vaccine                  6
flu shot  vaccine                  7
_
```

```
LIST OF COVID LABORATORIES:........

laboratory name                    priority
_____________________________________________
Clinical management of SARI           1
IHSTI                                 2
CNRS research laboratory              3
LDV USP LABS                          4
Applikon biotechnology                5
kakagowa laboratories                 6
gwangju institue of medicine          7
_
```

```
LIST OF COVID VACCINE MANUFACTURERS:........

manufacturer name                    priority
_____________________________________________
gen target inc                         1
pharma lonza                           2
eurofins                               3
astrazeneca                            4
virtuvax                               5
barr labs inc                          6
dynavax technologies                   7
```

```
LIST OF COVID VACCINE TESTING SUBJECTS:........

subject name                      priority
_____________________________________________
hamster                             1
pig                                 2
rabbit                              3
rat                                 4
lizard                              5
dog                                 6
human                               7
```

```
********************** Delete Menu **********************
Enter your choice
1. Delete covid infections record
2. Return to main menu
3. EXIT
```

```
********** COVID Infection Records **********
Choose from the following records to edit
1. world covid record
2. USA covid record
3. India covid record
4. France covid record
5. Brazil covid record
6. Netherlands covid record
7. Japan covid record
8. South Korea covid record
9. Return to main menu
10. EXIT
```

```
********** World Wide COVID Infections **********

cases  |  recovered |  active | serious | deaths |  population |
-------------------------------------------------------------
62004949 | 42798107 | 17757534 | 105249 | 1449308 | 3533463054 |

1 field was deleted from the queue

********** World Wide COVID Infections **********

 recovered |  active | serious | deaths |  population |
-----------------------------------------------------
42798107 | 17757534 | 105249 | 1449308 | 3533463054 |
```

```
********** USA COVID Infections **********

cases  |  recovered |  active | serious | deaths |  population |
-------------------------------------------------------------
10182818  |  3497817  |  18480  |  6441744  |  243257  |  331690410  |

1 field was deleted from the queue

********** USA COVID Infections **********

 recovered |  active | serious | deaths |  population |
-----------------------------------------------------
3497817  |  18480  |  6441744  |  243257  |  331690410  |
```

```
****************** India COVID Infections ***************

cases  |  recovered |  active | serious | deaths |  population |
-------------------------------------------------------------
850774  |  512624  |  8944  |  7868968  |  126162  |  1384789754  |

1 field was deleted from the queue

****************** India COVID Infections ***************

 recovered |  active | serious | deaths |  population |
-----------------------------------------------------------
512624  |  8944  |  7868968  |  126162  |  1384789754  |  _
```

```
****************** France COVID Infections ***************

cases  |  recovered |  active | serious | deaths |  population |
-------------------------------------------------------------
1748705  |  1580598  |  4421  |  127938  |  40169  |  65324901  |

1 field was deleted from the queue

****************** France COVID Infections ***************

 recovered |  active | serious | deaths |  population |
-----------------------------------------------------------
1580598  |  4421  |  127938  |  40169  |  65324901  |
```

```
****************** Brazil COVID Infections ***************

cases  |  recovered |  active | serious | deaths |  population |
-------------------------------------------------------------
5653561  |  426931  |  8318  |  5064344  |  162286  |  213094038  |

1 field deleted from the queue

****************** Brazil COVID Infections ***************

 recovered |  active | serious | deaths |  population |
-----------------------------------------------------------
426931  |  8318  |  5064344  |  162286  |  213094038  |
```

```
****************** Netherlands COVID Infections ***************

cases  |  recovered |  active | serious | deaths |  population |
-------------------------------------------------------------------
404401   |  2000   |  630   |  300205   |  7690   |  17148361   |

1 field deleted from the queue

****************** Netherlands COVID Infections ***************

 recovered |  active | serious | deaths |  population |
-----------------------------------------------------------
2000   |  630   |  300205   |  7690   |  17148361   |
```

```
****************** Japan COVID Infections ***************

cases  |  recovered |  active | serious | deaths |  population |
-------------------------------------------------------------------
105941   |   7644   |   194   |   96461   |   1809   |   126337911   |

1 field deleted from the queue

****************** Japan COVID Infections ***************

 recovered |  active | serious | deaths |  population |
-----------------------------------------------------------
7644   |   194   |   96461   |   1809   |   126337911   |
```

```
****************** South Korea COVID Infections ***************

cases  |  recovered |  active | serious | deaths |  population |
-------------------------------------------------------------------
27427   |   1981   |   58   |   24968   |   478   |   51284907   |

1 field deleted from the queue

****************** South Korea COVID Infections ***************

 recovered |  active | serious | deaths |  population |
-----------------------------------------------------------
1981   |   58   |   24968   |   478   |   51284907   |
```

ALGORITHM:

Declare all these as global variables

```
struct world *front=NULL,*rear=NULL; //world
struct world *front1=NULL,*rear1=NULL; //usa
struct world *front2=NULL,*rear2=NULL; //india
struct world *front3=NULL,*rear3=NULL; //france
struct world *front4=NULL,*rear4=NULL; //brazil
struct world *front5=NULL,*rear5=NULL; //netherlands
struct world *front6=NULL,*rear6=NULL; //japan
struct world *front7=NULL,*rear7=NULL; //south korea
struct vacc *frontq1=NULL,*rearq1=NULL; //vaccine names
struct vacc *frontq2=NULL,*rearq2=NULL; //laboratory names
struct vacc *frontq3=NULL,*rearq3=NULL; //manufacturer names
struct vacc *frontq4=NULL,*rearq4=NULL; //testing subjects
int i1=1;//world
int i2=1;//usa
int i3=1;//india
int i4=1;//france
int i5=1;//brazil
int i6=1;//netherlands
int i7=1;//japan
int i8=1;//south korea
```

Creating a structure called as vacc and world

```
typedef struct vacc
{
 char name[30];
 int p;
 struct node *link;
}vacc;

typedef struct world
{
 unsigned long int data;
 struct world *link;
}world;
```

These are the total functions used in the program :-

```
void worldcases();   //queue using link list
void displayworld(); //
void deleteworld();
void usacases();     //
void displayusa();   //
void deleteusa();
void indiacases();   //
void displayindia(); //
void deleteindia();
void francecases();  //
void displayfrance();//
void deletefrance();
void brazilcases();  //
void displaybrazil();//
void deletebrazil();
void netherlandscases();//
void displaynetherlands();//
void deletenetherlands();
void japancases();   //
void displayjapan();  //
void deletejapan();
void skoreacases();   //
void displayskorea(); //
void deleteskorea();
void vacc_names();    // priority queue using linked list
void displayvacc_name();
void lab_names();     //
void displaylab_name();  //
void man_names();       //
void displayman_name(); //
void testing_sub();
void display_sub();
```

In main function :

Step 1: START.

Step 2: Declare ch as integer.

Step 3: Use for loop for(; ;)
Inside the for loop perform the following code.

Step 4 : Print " MAIN MENU " on the screen and let the user enter their choice
1. Display Menu
2. Delete menu
3. EXIT

Step 5 : Read the value entered by the user in ‘ch’.
Step 6 : Use switch case

switch(ch)
case 1:
Print “DISPLAY MENU ” on the screen and let the user enter their choice
1. COVID Infection Cases information
2. COVID vaccination information
3. Return to Main Menu
4. EXIT
Read the value entered by the user in ‘ch’.

Again use switch case for the case 1 that is COVID Infection Cases information
switch(ch)
Print “COVID Infection Cases information” on the screen and let the user enter their choice
1. Display Total World Wide cases
2. Display info of any Specific country
3. Return to Main Menu
4. EXIT
Read the value entered by the user in ‘ch’.

Again use switch case
switch(ch)
case 1:
worldcases();
break;

case 2:
Print “ Country Wise COVID Infections ” on the screen and let the user enter their choice
1. USA
2. India
3. France
4. Brazil
5. Netherlands
6. Japan
7. South Korea
8. Return to main menu
9. EXIT
Read the value entered by the user in ‘ch’.

Again use switch case
switch(ch)
case 1://usa
usacases();
break;
case 2://india
indiacases();
break;

```
case 3://france
            francecases();
            break;
case 4://brazil
           brazilcases();
           break;
case 5://netherlands
            netherlandscases();
            break;
case 6://japan
            japancases();
            break;
case 7://south korea
            skoreacases();
            break;
 case 8:
            break;
case 9:
             exit(0);

default:
                Print " Invalid Choice Try Again "
                break;

break;

break;
```

case 2://vaccine info
Print " COVID Vaccination Info Menu " on the screen and let the user enter their choice
1. Display list of vaccine names
2. Display list of laboratory names involved in covid vaccine creation
3. Display list of manufacturer names involved in vaccine creation
4. Display list of testing subjects used in vaccine creation
5. Return to main menu
6. EXIT
Read the value entered by the user in 'ch'.

Again use switch case
switch(ch)

```
case 1:                         //vaccine names
            vacc_names();
           break;
case 2:                         //lab names
          lab_names();
           break;
case 3:                         //manufacturer names
           man_names();
           break;
```

```
  case 4:                          //testing subjects
              testing_sub();
              break;
  case 5:
            break;
  case 6:
            exit(0);
             break;
  default:
            Print  " Invalid Choice Try Again "
             break;
break;

    case 3:                        //return to main menu
                    break;

    case 4:
                     exit(0);

     default:
                    Print " Invalid choice Try Again "
                    break;
    break;

     case 2 :                      // edit menu
     Print " Delete menu " on the screen and let the user enter their  choice
     1. Delete covid infections record
     2. Return to main menu
     3. EXIT
     Read the value entered by the user in 'ch'.

   Again use switch case
   switch(ch)

  case 1 : //covid infections menu
      Print " COVID Infection Records " on the screen and let the user enter their     choice to
      edit
      1. world covid record
      2. USA covid record
      3. India covid record
      4. France covid record
      5. Brazil covid record
      6. Netherlands covid record
      7. Japan covid record
      8. South Korea covid record
      9. Return to main menu
      10. EXIT

  Read the value entered by the user in 'ch'.
```

```
Again use switch case
    switch(ch)

case 1:                                  //world
            worldcases();
            deleteworld();
           break;

 case 2:                                 //usa
            usacases();
            deleteusa();
            break;

case 3:                                  //india
            indiacases();
            deleteindia();
           break;

 case 4:                                 //france
             francecases();
            deletefrance();
             break;

   case 5:                               //brazil
                brazilcases();
                deletebrazil();
               break;

  case 6:                                //netherlands
                netherlandscases();
                 deletenetherlands();
                break;

   case 7:                               //japan
                 japancases();
                 deletejapan();
                 break;

   case 8:                                //south korea
                 skoreacases();
                 deleteskorea();
                  break;
   case 9:
                 break;

  case 10:
                exit(0);
```

default:
Print " Invalid Choice Try Again "
Break;

case 2:
break;

case 3:
exit(0);

default:
Print " invalid choice try again "
break;

break

case 3:
exit(0);
break;

default:
Print " Invalid choice Try again "
break;

Step 7 : STOP.

Create a function called as displayvacc_name()

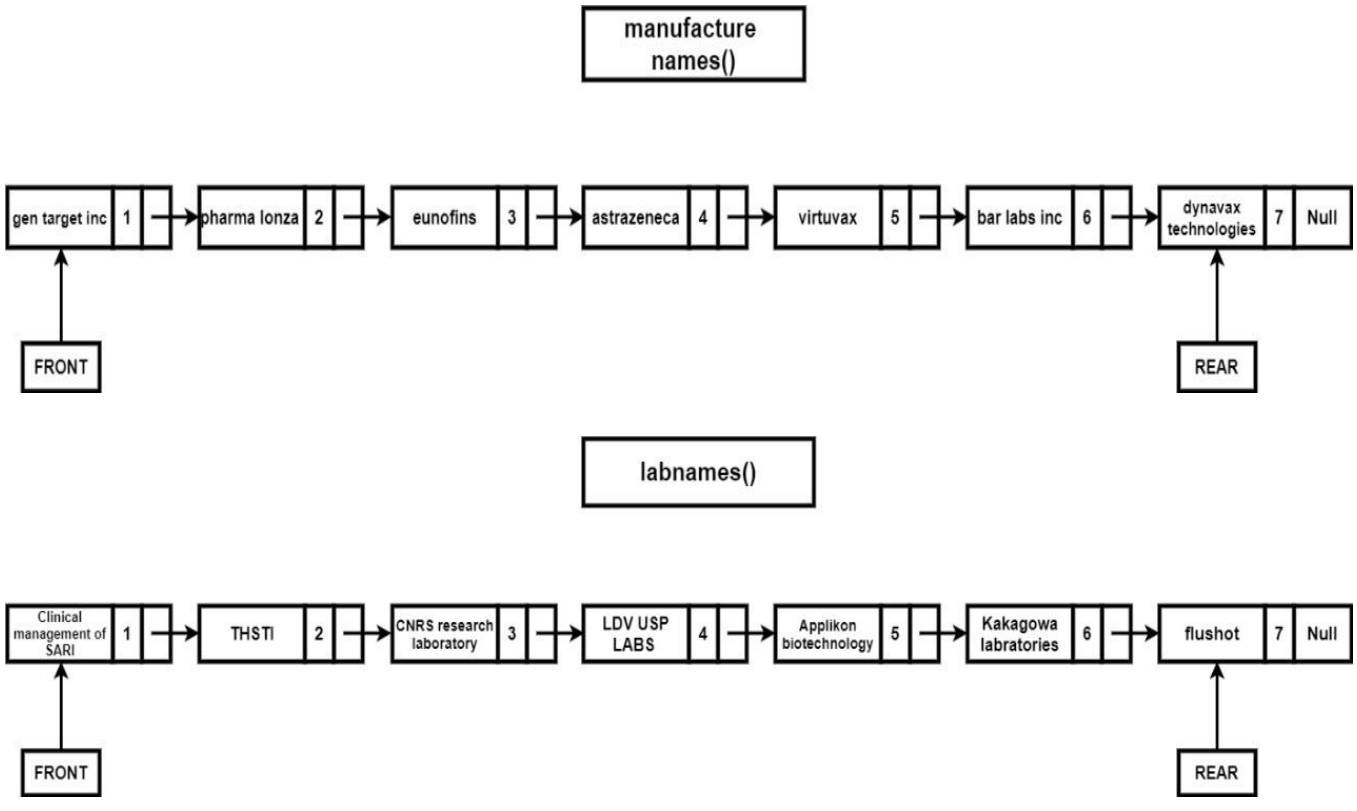
Step 1 : Declare *temp as struct vacc
Step 2 : temp=front1
Step 3 : Print " LIST OF COVID VACCINES:..... "
Print " vaccine name priority "
Print " _____ "
Step 4 : Use while loop until temp !=NULL
Print " %s vaccine %d\n",temp->name,temp->p "
temp=temp->link;
Step 5 : End

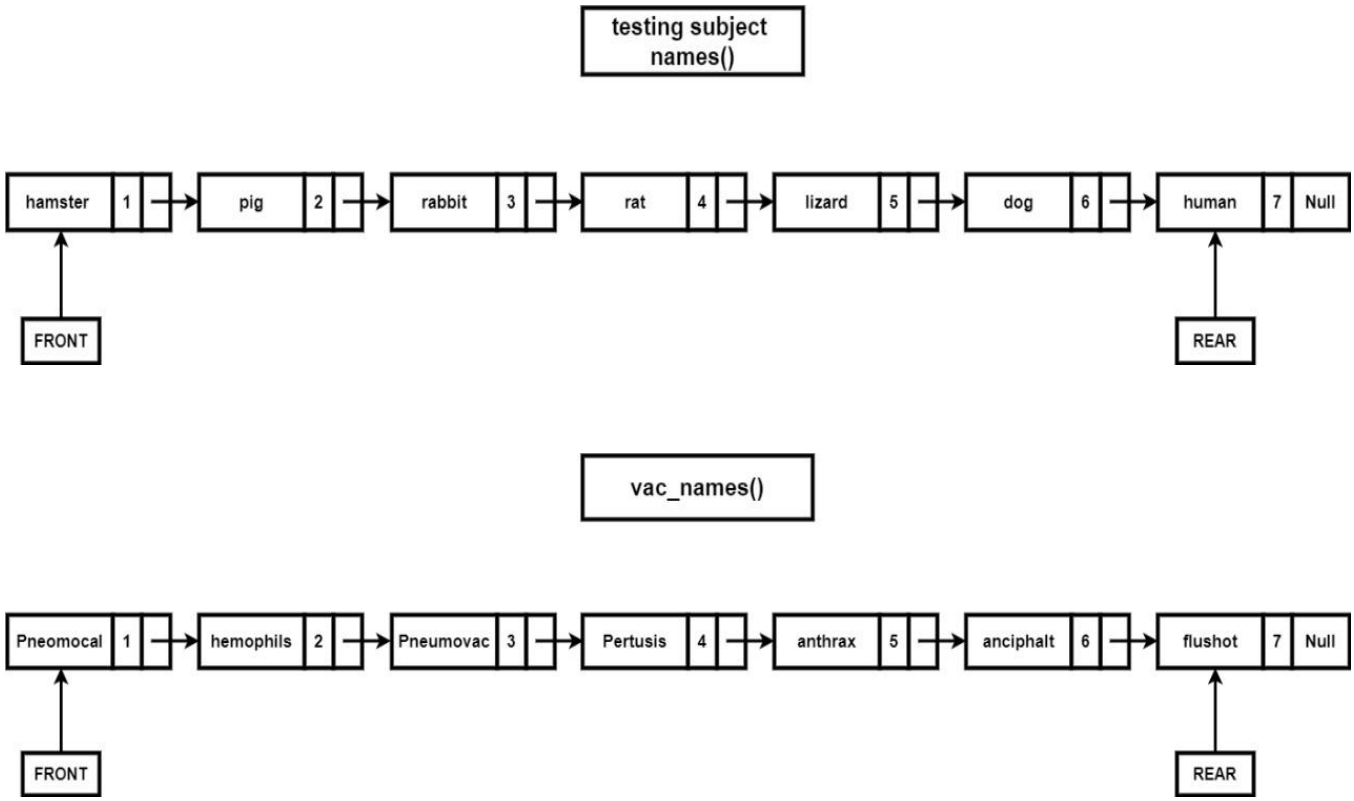
Similarly create functions for displaylab_name() , displayman_name() , display_sub().

Create a function called as vacc_names()

Step 1 : Declare * ptr as vacc.
Step 2 : Allocate memory for first node
frontq1=(vacc*)malloc(sizeof(vacc));
Step 3 : Inserting values in the first node
strcpy(frontq1->name,"Pneumocal");
frontq1->p=1;
frontq1->link=NULL;
Step 4 : rearq1=frontq1;
Step 5 : Allocate memory for new node
ptr=(vacc*)malloc(sizeof(vacc));
Step 6 : Inserting values in the node
strcpy(ptr->name,"hemophils");
ptr->p=2;
ptr->link=NULL;
Step 7 : rearq1->link=ptr;
rearq1=ptr;
Step 8 : Similarly insert the remaining records
Step 9 : At the last call the displayvacc_name() to display the records .
Step 10 : End.

With respect to above algorithm , create similar functions named as lab_names() , man_names(), testing_sub() and call displaylab_name(), displayman_name() , display_sub() respectively with different records in every function.



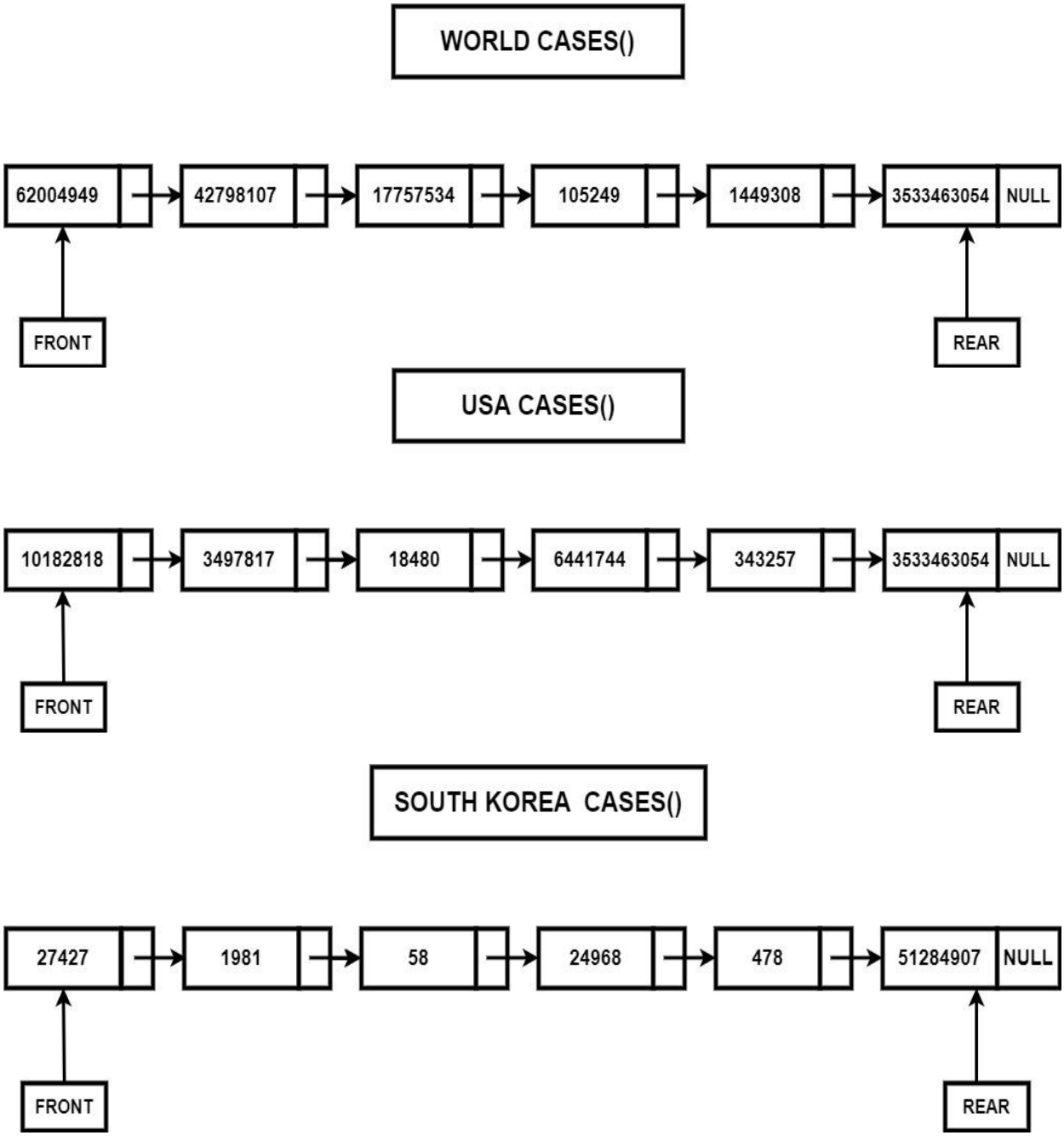


Create a function called as worldcases()

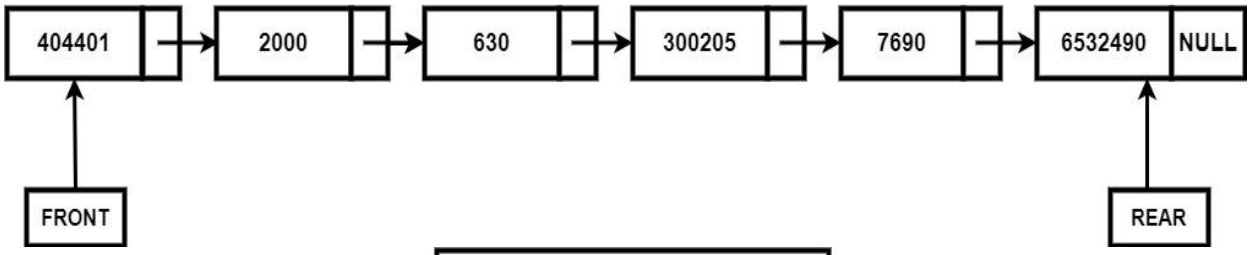
Step 1 : Declare * ptr as world i.e(structure),
a[6]={62004949,42798107,17757534,105249,1449308,7828430350} as unsigned long int and i as int
Step 2 : Allocate memory for first node
front=(world*)malloc(sizeof(world));
Step 3 : Inserting values of the first node
front->data=a[0];
front->link=NULL;
Step 4 : rear=front;
Step 5 : Check if (i1==1) then goto step 6 else goto step 15
Step 6 : Use for loop for(i=1;i<6;i++)
Step 6a : ptr=(world*)malloc(sizeof(world)); //to allocate memory for new node
ptr->data=a[i]; //for inserting values of a particular index value of a
ptr->link=NULL;
rear->link=ptr;
rear=ptr;
step 7 : Check else if (i1==2) then goto step 8 else goto step 15
Step 8 : Use for loop for(i=2;i<6;i++)
Repeat step 6a.
Step 9 : Check else if (i1==3) then goto step 10 else goto step 15
Step 10 : Use for loop for(i=3;i<6;i++)
Repeat step 6a.
Step 11 : Check else if (i1==4) then goto step 12 else goto step 15
Step 12 : Use for loop for(i=4;i<6;i++)
Repeat step 6a.

Step 13 : Check else if (i1==5) then goto step 14 else goto step 15
Step 14 : Use for loop for(i=5;i<6;i++)
Repeat step 6a.
Step 15 : Call displayworld()
Step 16 : End .

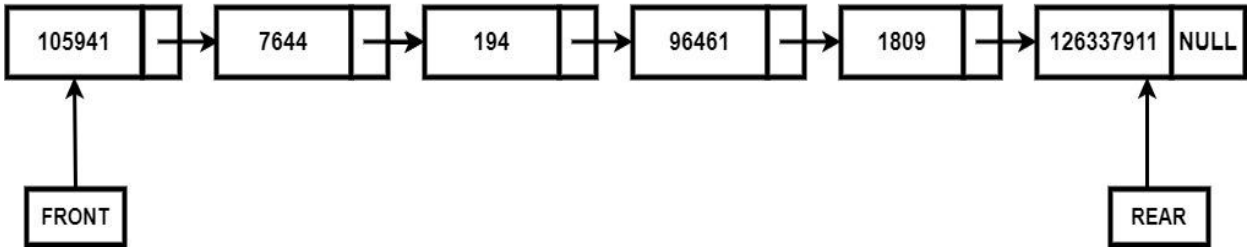
With respect to above algorithm , create similar functions named as usacases(), indiacases() , francecases() , brazilcases() , netherlandscases(), japancases() , skoreacases() respectively with different records in every function .



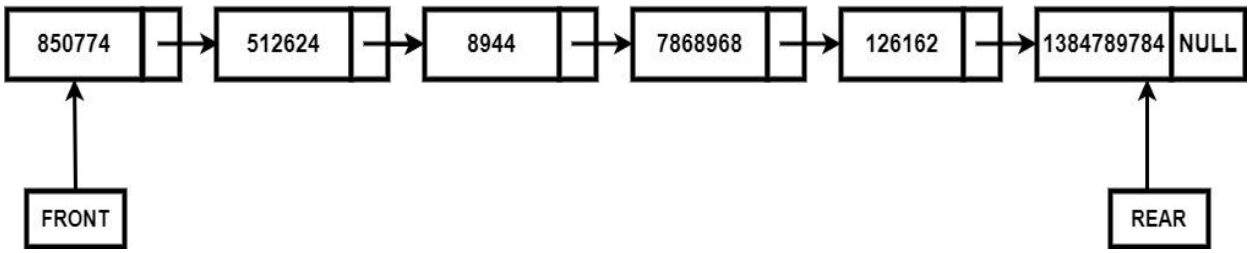
NETHERLANDS CASES()



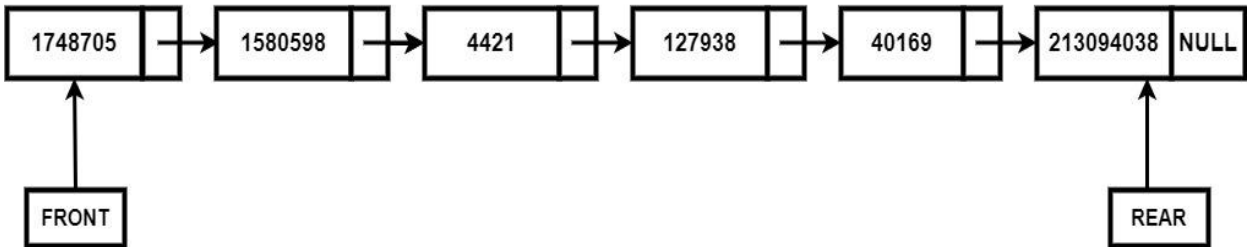
JAPAN CASES()

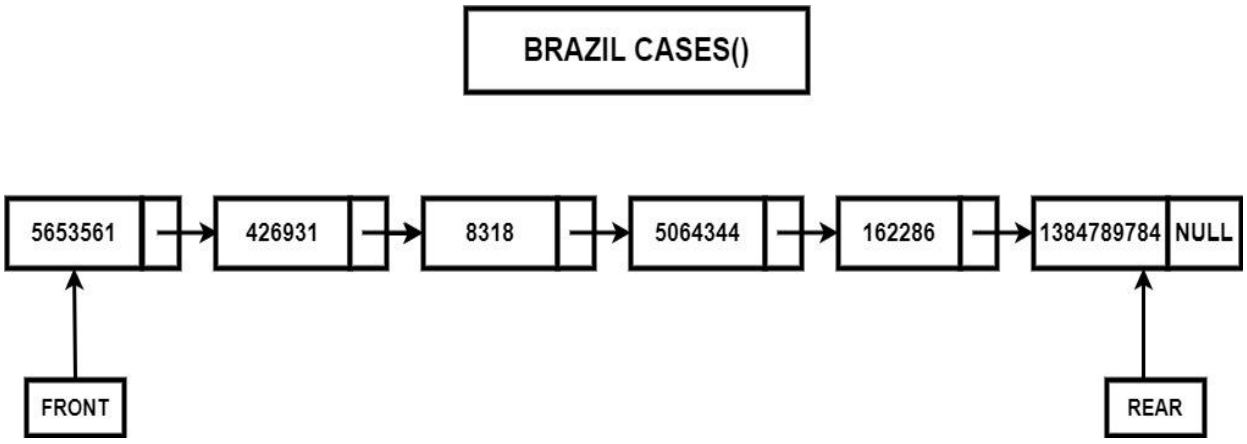


INDIA CASES()



FRANCE CASES()





Create a function called as displayworld ()

Step 1 : Declare * temp as world i.e(structure)
Step 2 : temp=front
Step 3 : Check if (i1==1) then goto step 4
Step 4 : Print " World Wide COVID Infections "
Step 5 : Print "cases | recovered | active | serious | deaths | population | "
Print "---------- "
Step 6: Use while loop until temp!=NULL
Print ("%lu | ",temp->data)
temp=temp->link
Step 7 : Check else if (i1==2) then goto step 8
Step 8 : Repeat step 4
Print " recovered | active | serious | deaths | population | "
Print "---------- "
Repeat step 6
Step 9 : Check else if (i1==3) then goto step 10
Step 10 : Repeat step 4
Print " active | serious | deaths | population | "
Print " ---------- "
Repeat step 6
Step 11 : Check else if (i1==4) then goto step 12
Step 12 : Repeat step 4
Print " serious | deaths | population | "
Print " ---------- "
Repeat step 6
Step 13 : Check else if (i1==5) then goto step 14
Step 14 : Repeat step 4
Print " deaths | population | "
Print " ---------- "

Repeat step 6
Step 15 : Check else if (i1==6) then goto step 16
Step 16 : Repeat step 4
Print " population | "
Print " ---------- "
Repeat step 6
Step 17 : Print " Queue is empty "

Step 18 : End .

With respect to above algorithm, create similar functions named as displayusa() , displayindia() , displayfrance() , displaybrazil() , displaynetherlands() , displayjapan() , displayskorea() .

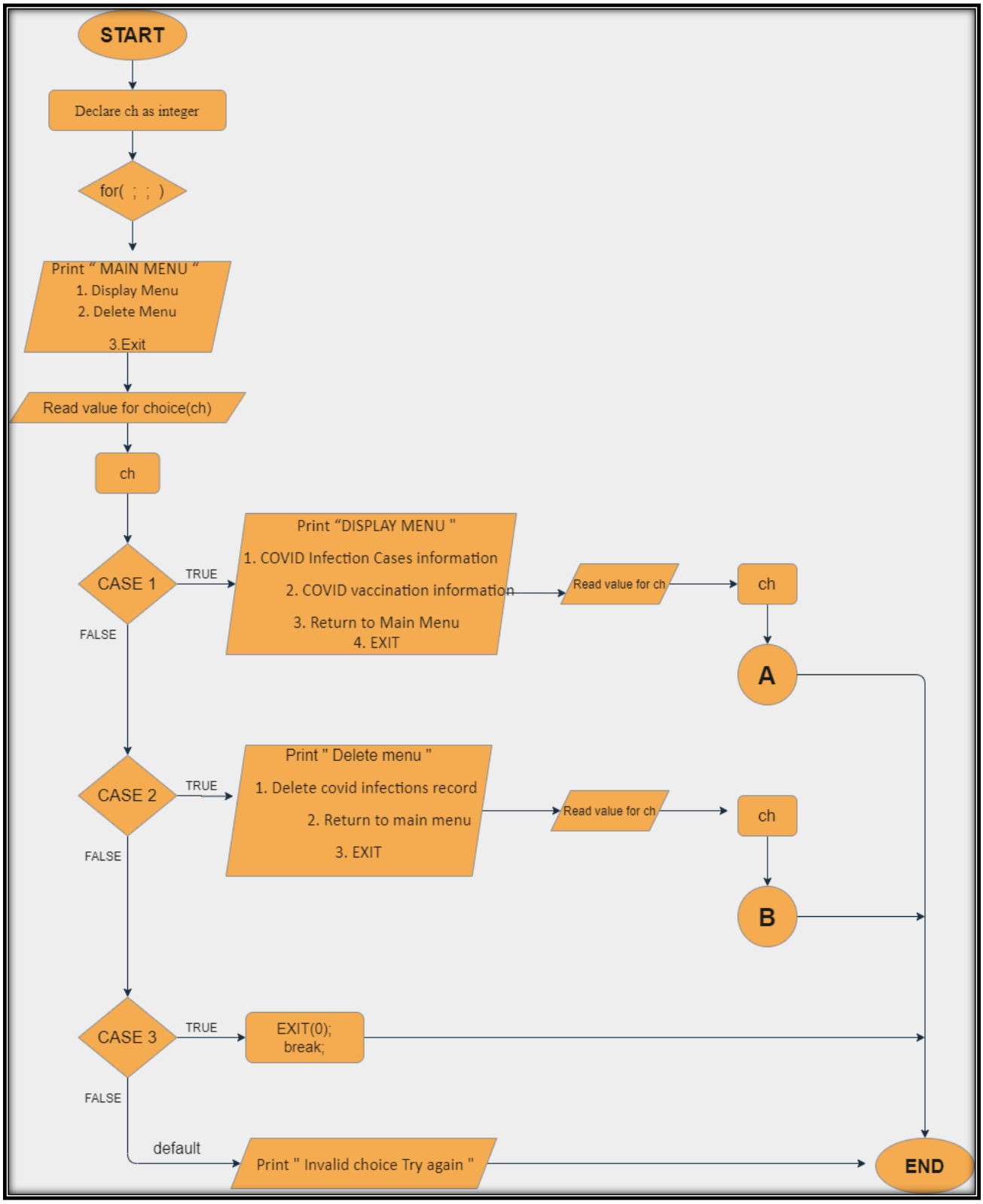
Create a function called as deleteworld ()

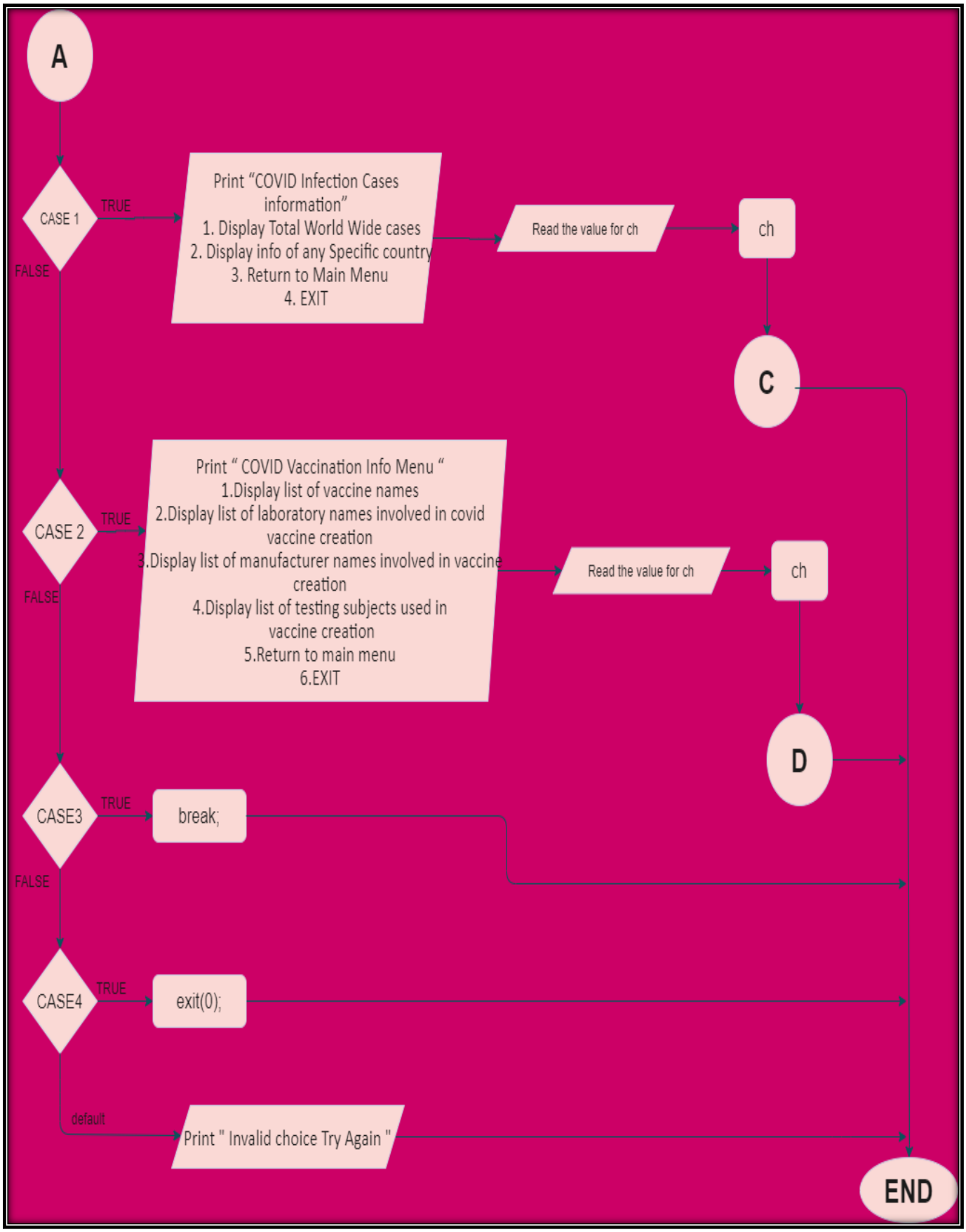
Step 1 : Declare * temp as world i.e(structure)
Step 2 : temp=front //
Step 3 : Front will start pointing to the next node i.e front=front->link
Step 4 : Print ("\n\n%d field was deleted from the queue\n\n",i1)
Step 5 : Increment the vvalue of i1 by 1 i.e i1++
Step 6 : Free the memory location of temp i.e free(temp)
Step 7 : Call the displayworld()
Step 8 : End.

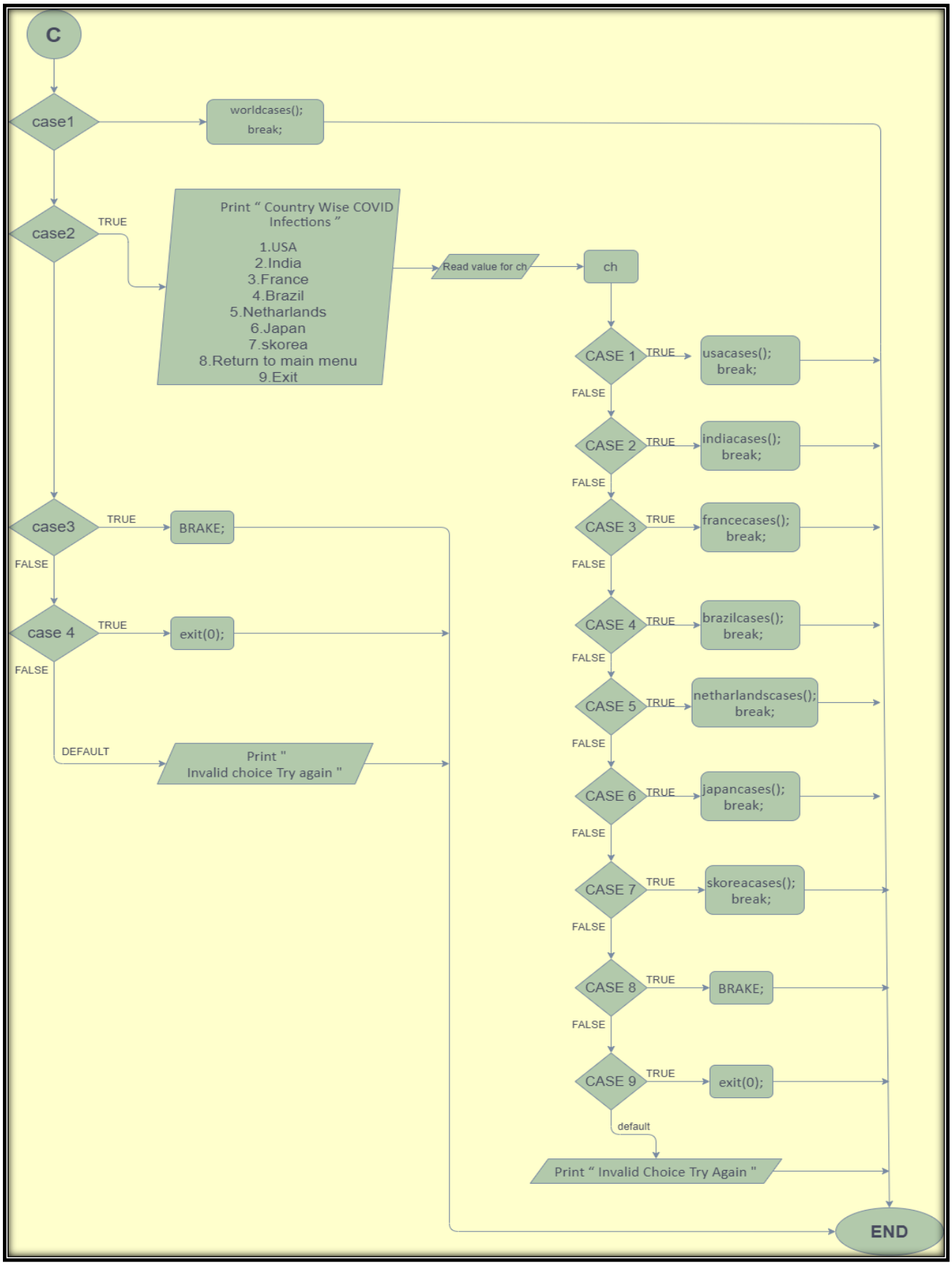
With respect to above algorithm, create similar functions named as deleteusa() , deleteindia() , deletefrance() , deletebrazil() , deletenetherlands() , deletejapan() , deleteskorea() .

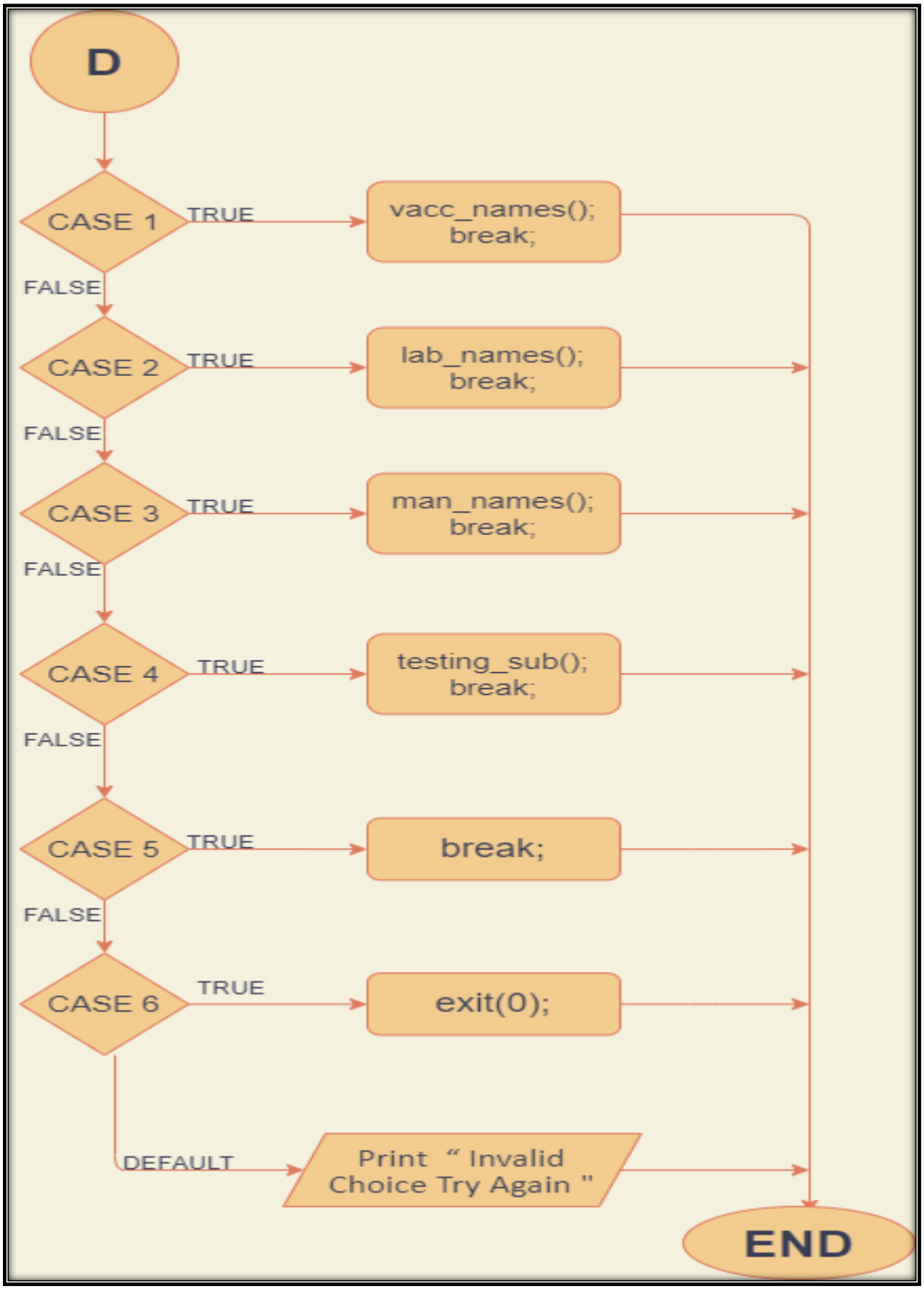
FLOWCHART:

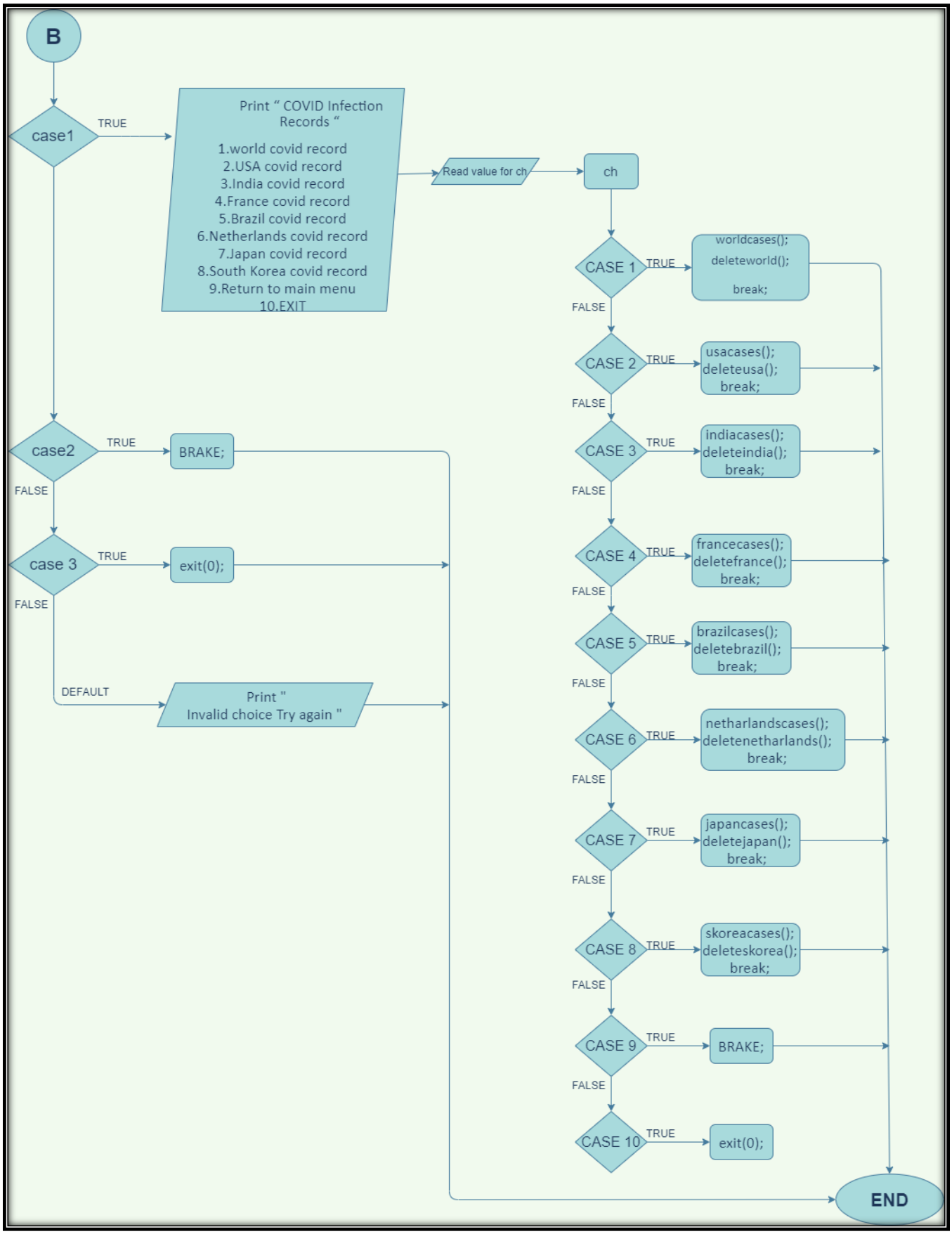
Flowchart for main function() :



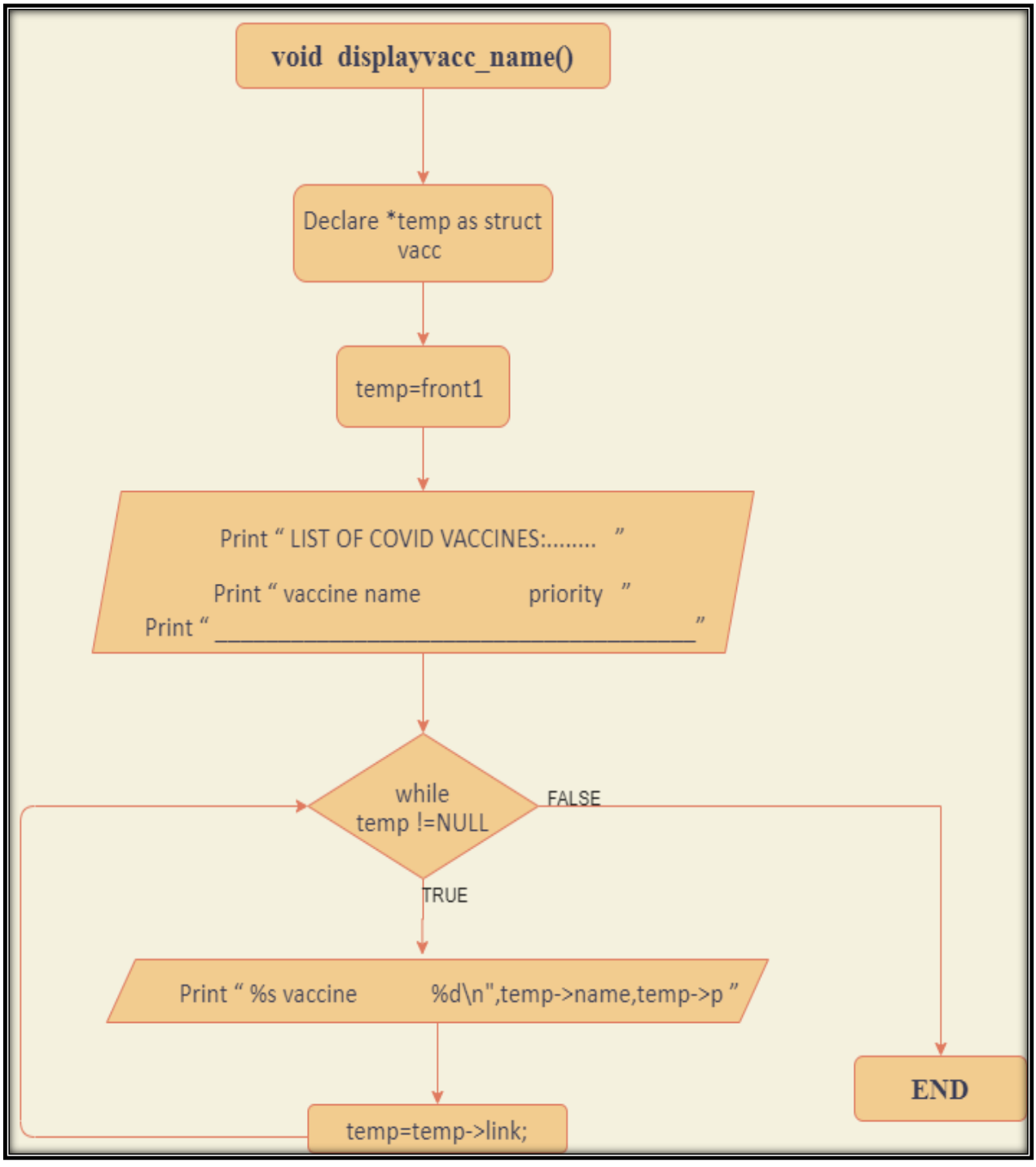

A
CASE 1
TRUE
FALSE
Print "COVID Infection Cases information"
1. Display Total World Wide cases
2. Display info of any Specific country
3. Return to Main Menu
4. EXIT
Read the value for ch
ch
C
CASE 2
TRUE
FALSE
Print " COVID Vaccination Info Menu "
1.Display list of vaccine names
2.Display list of laboratory names involved in covid vaccine creation
3.Display list of manufacturer names involved in vaccine creation
4.Display list of testing subjects used in vaccine creation
5.Return to main menu
6.EXIT
Read the value for ch
ch
D
CASE3
TRUE
break;
FALSE
CASE4
TRUE
exit(0);
default
Print " Invalid choice Try Again "
END


C
case1
worldcases();
break;
case2
TRUE
Print " Country Wise COVID Infections "
1.USA
2.India
3.France
4.Brazil
5.Netharlands
6.Japan
7.skorea
8.Return to main menu
9.Exit
Read value for ch
ch
CASE 1
TRUE
usacases();
break;
FALSE
CASE 2
TRUE
indiacases();
break;
FALSE
CASE 3
TRUE
francecases();
break;
FALSE
CASE 4
TRUE
brazilcases();
break;
FALSE
CASE 5
TRUE
netharlandscases();
break;
FALSE
CASE 6
TRUE
japancases();
break;
FALSE
CASE 7
TRUE
skoreacases();
break;
FALSE
CASE 8
TRUE
BRAKE;
FALSE
CASE 9
TRUE
exit(0);
default
Print " Invalid Choice Try Again "
case3
TRUE
BRAKE;
FALSE
case 4
TRUE
exit(0);
FALSE
DEFAULT
Print "
Invalid choice Try again "
END


D
CASE 1
TRUE
vacc_names();
break;
FALSE
CASE 2
TRUE
lab_names();
break;
FALSE
CASE 3
TRUE
man_names();
break;
FALSE
CASE 4
TRUE
testing_sub();
break;
FALSE
CASE 5
TRUE
break;
FALSE
CASE 6
TRUE
exit(0);
DEFAULT
Print " Invalid
Choice Try Again "
END

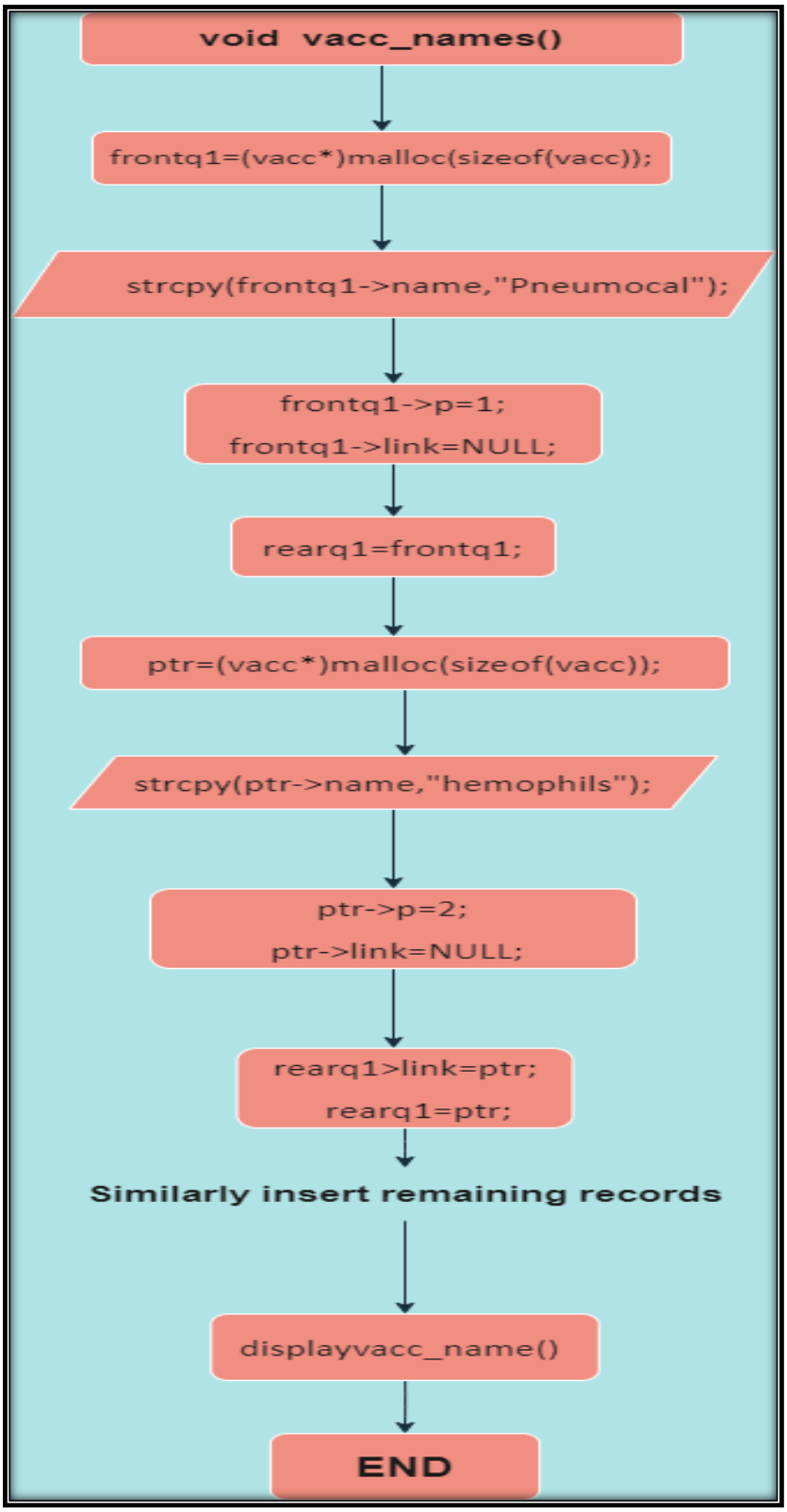

B
case1
TRUE
Print " COVID Infection Records "
1.world covid record
2.USA covid record
3.India covid record
4.France covid record
5.Brazil covid record
6.Netherlands covid record
7.Japan covid record
8.South Korea covid record
9.Return to main menu
10.EXIT
Read value for ch
ch
CASE 1
TRUE
worldcases();
deleteworld();
break;
FALSE
CASE 2
TRUE
usacases();
deleteusa();
break;
FALSE
CASE 3
TRUE
indiacases();
deleteindia();
break;
FALSE
CASE 4
TRUE
francecases();
deletefrance();
break;
FALSE
CASE 5
TRUE
brazilcases();
deletebrazil();
break;
FALSE
CASE 6
TRUE
netharlandscases();
deletenetharlands();
break;
FALSE
CASE 7
TRUE
japancases();
deletejapan();
break;
FALSE
CASE 8
TRUE
skoreacases();
deleteskorea();
break;
FALSE
CASE 9
TRUE
BRAKE;
FALSE
CASE 10
TRUE
exit(0);
case2
TRUE
BRAKE;
FALSE
case 3
TRUE
exit(0);
FALSE
DEFAULT
Print "
Invalid choice Try again "
END

Flowchart for function displayvacc_name() :



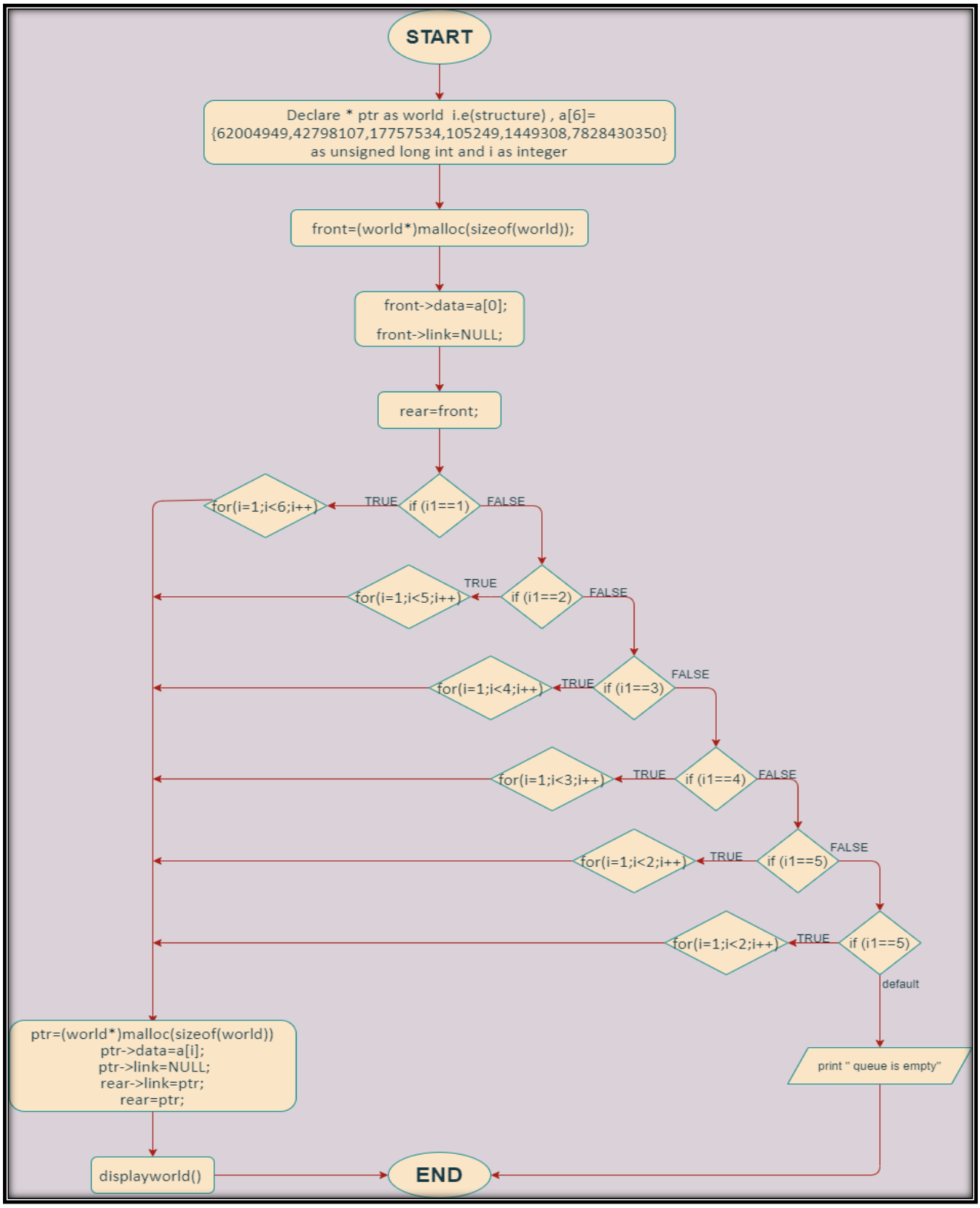
With respect to above flowchart, we can create flowchart for similar functions like displaylab_name() , displayman_name() , display_sub().

Flowchart for function vacc_names() :



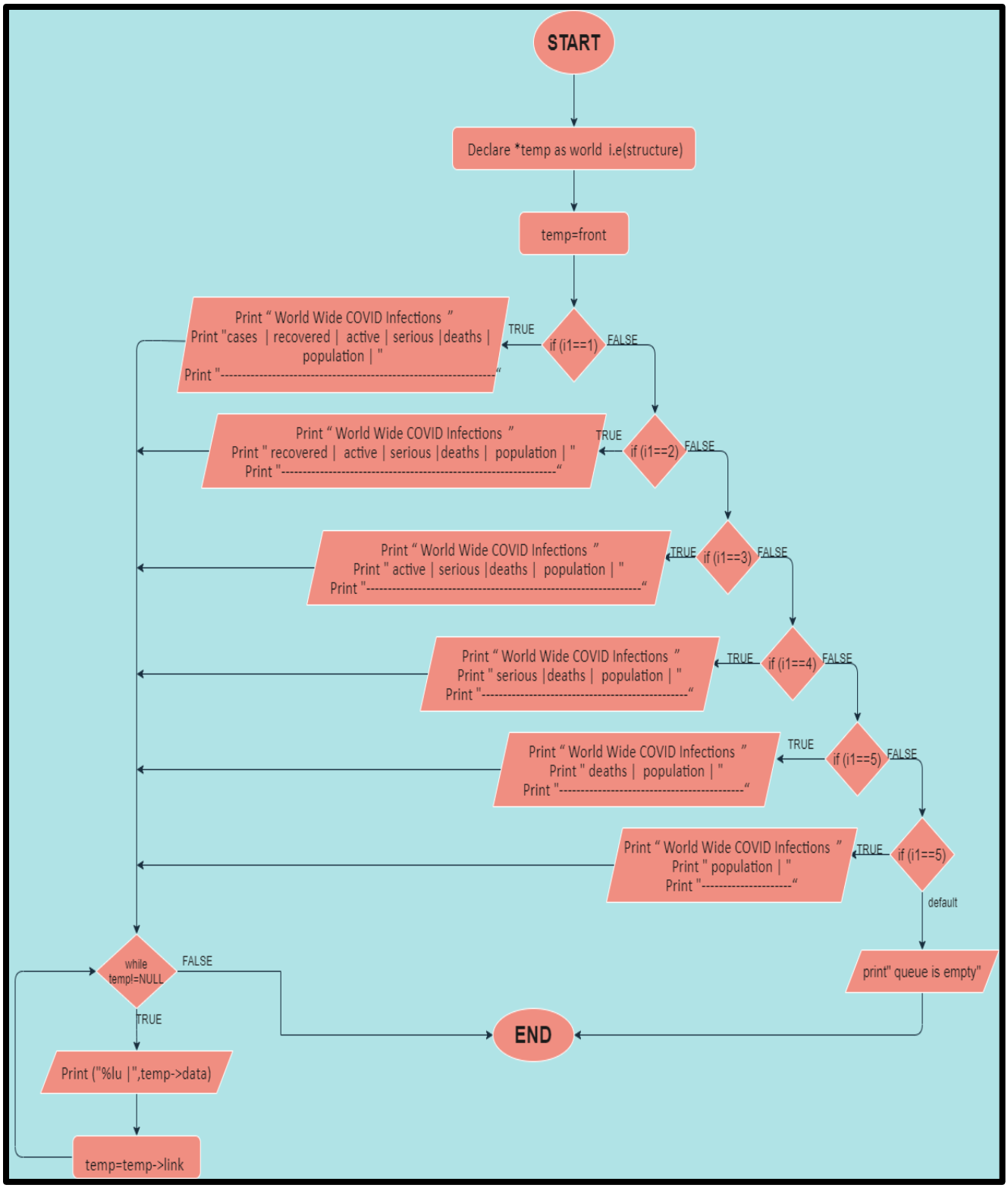
With respect to the given flowchart, we can create flowchart for similar functions named as lab_names( ), man_names( ), testing_sub( ) and call displaylab_name( ), displayman_name( ), display_sub( ) respectively with different records in every function .

Flowchart for function worldcases() :



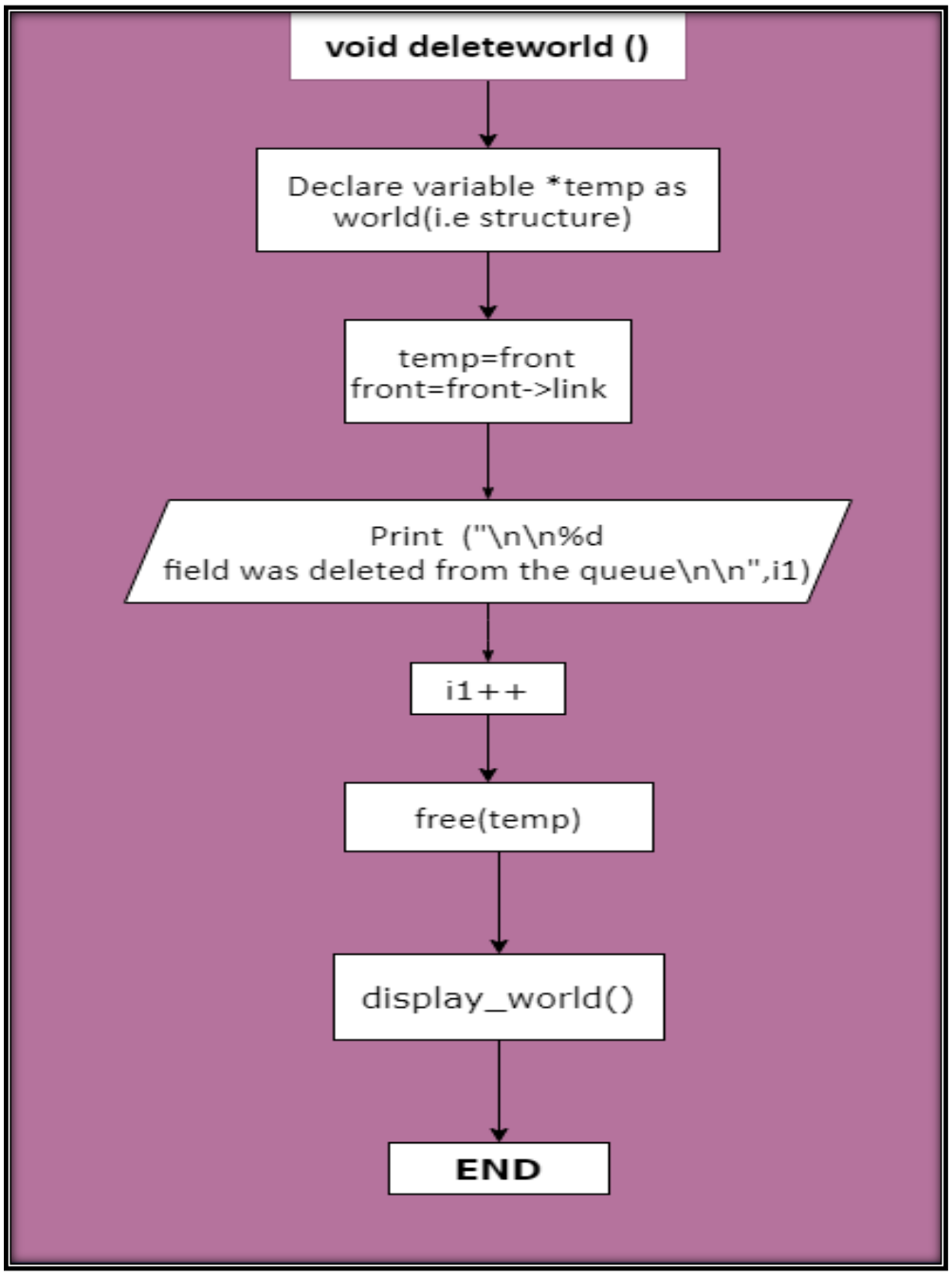
With respect to above flowchart , we can create flowchart for similar functions named as usacases( ), indiacases( ), francecases( ), brazilcases( ), netherlandscases( ) , japancases( ), skoreacases( ).

Flowchart for function displayworld ( ):



With respect to above flowchart, we can create similar flowchart for functions named as displayusa( ), displayindia( ), displayfrance( ), displaybrazil( ), displaynetherlands( ) ,displayjapan(), displayskorea( ).

Flowchart for function deleteworld( ):



With respect to above flowchart, we can create flowchart fof similar functions named as deleteusa( ), deleteindia( ), deletefrance( ), deletebrazil( ), deletenetherlands( ), deletejapan( ), deleteskorea( ) .