CSc 3320: Systems Programming

Spring 2021

Midterm 2: Total points = 100

Assigned: 11th Apr 2021, Sunday 11:59 PM **Submission Deadline: 18th Apr 2021, Sunday, 11.59 PM (No extensions. If your submission is not received by this time then it will NOT be accepted.)**

Submission instructions:

1. Create a Google doc for your submission.
2. Start your responses from page 2 of the document and copy these instructions on page 1.
3. Fill in your name, campus ID and panther # in the fields provided. If this information is missing TWO POINTS WILL BE DEDUCTED.
4. Keep this page 1 intact. If this *submissions instructions* page is missing in your submission TWO POINTS WILL BE DEDUCTED.
5. Start your responses to each QUESTION on a new page.
6. If you are being asked to write code copy the code into a separate txt file and submit that as well. The code should be executable. E.g. if asked for a C script then provide myfile.c so that we can execute that script. In your answer to the specific question, provide the steps on how to execute your file (like a ReadMe).
7. If you are being asked to test code or run specific commands or scripts, provide the evidence of your outputs through a screenshot and/or screen video-recordings and copy the same into the document.
8. Upon completion, download a .PDF version of the google doc document and submit the same along with all the supplementary files (videos, pictures, scripts etc).

Full Name: Hakan Gunerli

Campus ID: hgunerli1

Panther #: 002504797

Questions 1-3 are 20pts each. Question 4 is 40pts

All programs have to be well commented. Non commented programs will receive

0 points. Comments have to be easily comprehensible and concise.

1. Consider the array given below. Write a C program that must be able to sort the elements in the array. You must use pointers in your code to work with the arrays. The sort functionality must be implemented as a separate function named “sort_numeric()”

   Array for your evaluation

   [10, 0.25, -2342, 12123, 3.145435, 6, 6, 5.999, -2, -5, -109.56]

   If given user input A or a: sort in Ascending order

   If given user input D or d: sort in Descending order

2. Consider the list of names given below. Write a C program that will first create a string array that will contain this list and then sort the elements in the array as per alphabetical order. You must use pointers in your code to work with the arrays. The sort functionality must be implemented as a separate function named “sort_alphabetic()”. The program can be case insensitive (i.e. capital or small letters are treated the same).

   List for your evaluation

   Systems
   Programming
   Deep
   Learning
   Internet
   Things
   Robotics
   Course

   If given user input A or a: sort in alphabetical order (a comes first) If given user input D or d: sort in reverse alphabetical order(z comes first)

3. Repeat Question 1 or Question 2, considering that the number of elements can potentially increase. That is, the size of the array will be unknown at the start of

the program. Note that the requirement of using pointers still holds.

Show proof of evaluation of your program being able to work for more than 10 entries. Show 5 evaluation trials in your submission. You can pick any number of entries between 10 and 30 for your trials.

(Hint: *To solve this, use dynamic memory allocation, where you will NOT treat the input array as a known or finite size. Allocate memory space (e.g. malloc()) as and when the number of elements in the list increases).*

4. Using C programming and using Structures or Unions in your program, build a COVID vaccine registration form where any user can register by filling in their First Name, Last Name, Date of Birth (mm/dd/yyyy), Sex, Dose number (1 or 2), Date of previous dose, Type of vaccine (Pfizer, Moderna, Johnson&Johnson), Residential zipcode.

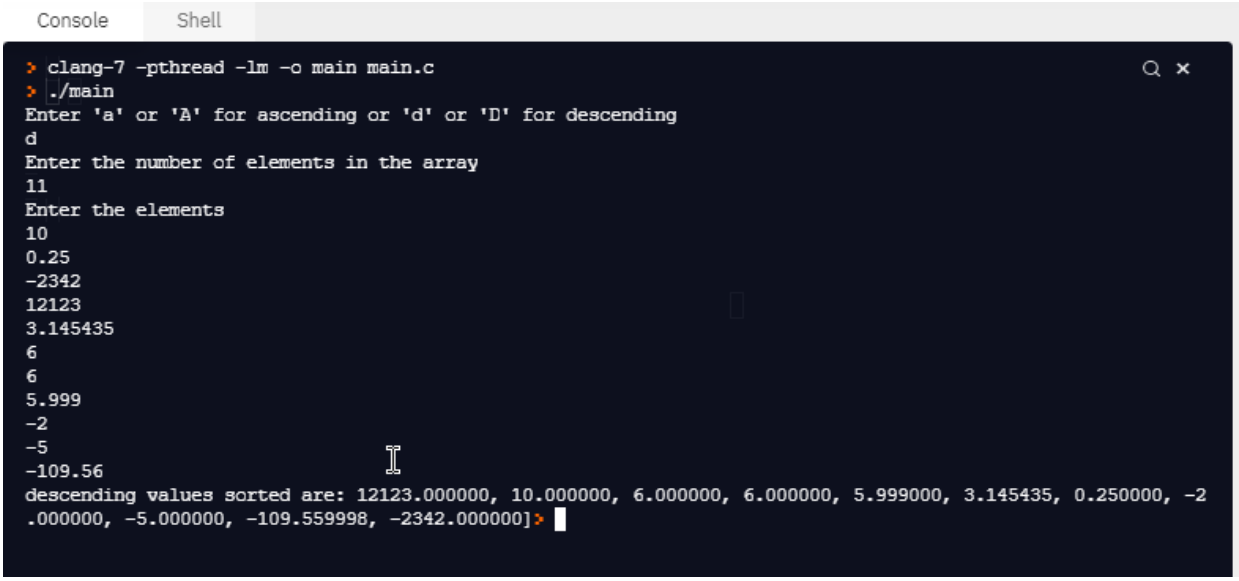
   Upon registration, the system must output a 8 letter alphanumeric code that will be unique to that user. The code is generated as <First letter of First Name><First Letter of Last Name><current age of user -as of registration date><First letter of Vaccine type><last 3 numbers of zipcode>

   Add functionality in your program such that it will display all the user's information on the screen (one item in each line).

   Show an evaluation trial for registering at least 10 users. For registration, ,for relevant questions, users must choose values based on the options provided (e.g. sex; options must be Male/Female/Do not wish to identify)

   (Hint: Write a program that contains main(), register(), generate_code() and retrieve() functions, at the least).

QUESTION 1



How to run it:

gcc question1.c && ./a.out

First mention whether you want ascending or descending order, then Input how many characters you will be adding.

```c
#include <stdio.h>

int sort_numeric(float *array, char input, int len);
```

```c
int sort_numeric(float *array, char input, int len)

{

        int first, second, i;

        double tmp;                         // temp variable will be used for
bubble sort later on, check L17

        if (input == 'a' || input == 'A') // if the input is A or A

        {



                for (first = 0; first < len; first++) // until the length of
the array is finished

                {



                        for (second = first + 1; second < len; second++)

                        {

                                if (*(array + first) >= *(array + second)) //
if array [the user input] plus first is larger than the second then using temp
switch their positions. like a bubble sort.

                                {
```

```
                                        tmp = *(array + first);
// compare-switch
                                        *(array + first) = *(array + second);
// compare-switch
                                        *(array + second) = tmp;
// compare-switch
                                }
                        }
                }
        }
        else if (input == 'd' || input == 'D')
        {
                for (first = 0; first < len; first++)
                {
                        for (second = first + 1; second < len; second++)
                        {
                                if (*(array + first) <= *(array + second)) //
if array [the user input] plus first is less than the second then using temp
```

```
switch their positions. like a bubble sort.

                                {




                                        tmp = *(array + first);
// compare-switch

                                        *(array + first) = *(array + second);
//compare-switch

                                        *(array + second) = tmp;
// compare-switch

                                }

                        }

                }

        }



        else

        {

                printf("option not available .\n"); // return an error if it's
malfunctioning

                return 0;
```

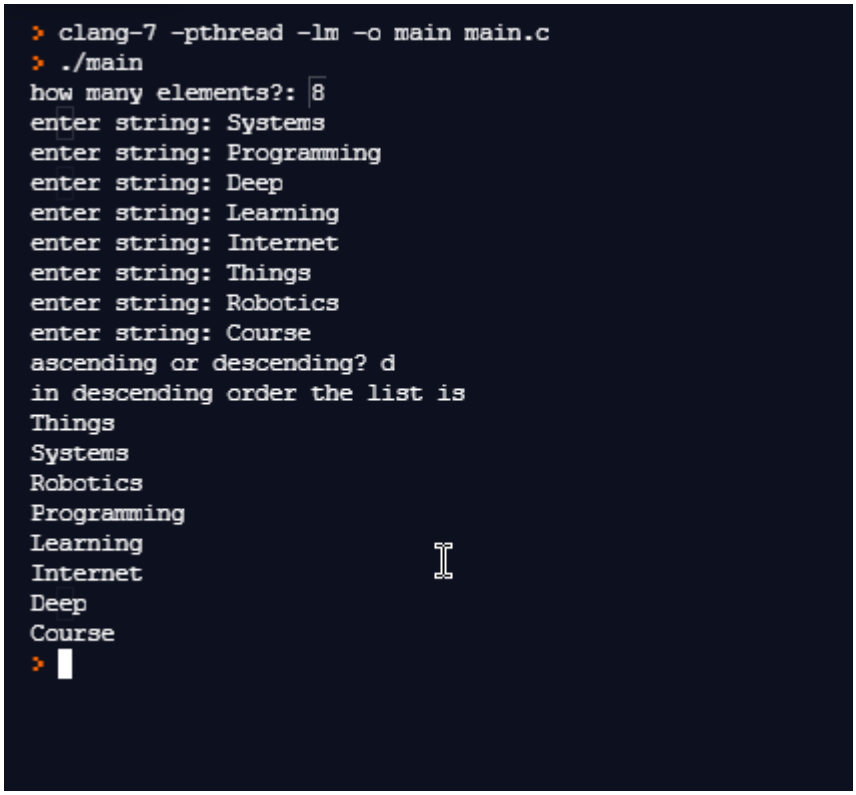
```
        }

        // just a little personal addition here, tell the user if their inputs
are ascending descending or an error.

        if (input == 'a' || input == 'A')

        {

                printf("ascending");

        }

        else if (input == 'd' || input == 'D')

        {

                printf("descending");

        }

        else

        {

                printf("error");

        }



        printf(" values sorted are: ");
```

```
        for (i = 0; i < len; i++)
        {
                if (i != len - 1)
                {
                        printf("%f, ", array[i]);
                }
                else
                {
                        printf("%f]", array[i]);
                }
        }
        return 0;
}

int main()
{
```

```
        char inp;

        int len, i;



        printf("Enter 'a' or 'A' for ascending or 'd' or 'D' for
descending\n"); // ask the user for their input

        scanf("%c", &inp);



        printf("Enter the number of elements in the array and enter the
elements one after the other:\n");

        scanf("%d", &len); // how many elements, which will be used for
iteration later on

        float arr[len];

        printf("Enter the elements\n");

        for (i = 0; i < len; i++) // ask the user for their input one by one

        {

                scanf("%f", &arr[i]); // scan all as float since we have some
that have decimal points on em.

        }

        sort_numeric(arr, inp, len); // pass that to the method
```

```
        return 0;

}
```

QUESTION 2



HOW TO RUN IT :

gcc question2.c && ./a.out

Enter how many elements you are planning to type, then start entering the amount of strings and the order you wish them to be. Then the program will make it ascending or descending accordingly.

```c
#include <stdio.h>

#include <string.h>

#include <ctype.h>

#include <stdlib.h>



void sort_alphabetic(char *string[]); // telling the compiler that i'll be
using this.



int main(void)

{

    int val;




    printf("how many elements?: ");

    scanf("%d", &val);

     char *list[val*100]; // {"Systems ", "Programming", "Deep", "Learning",
"Internet", "Things", "Robotics", "Course"};
```

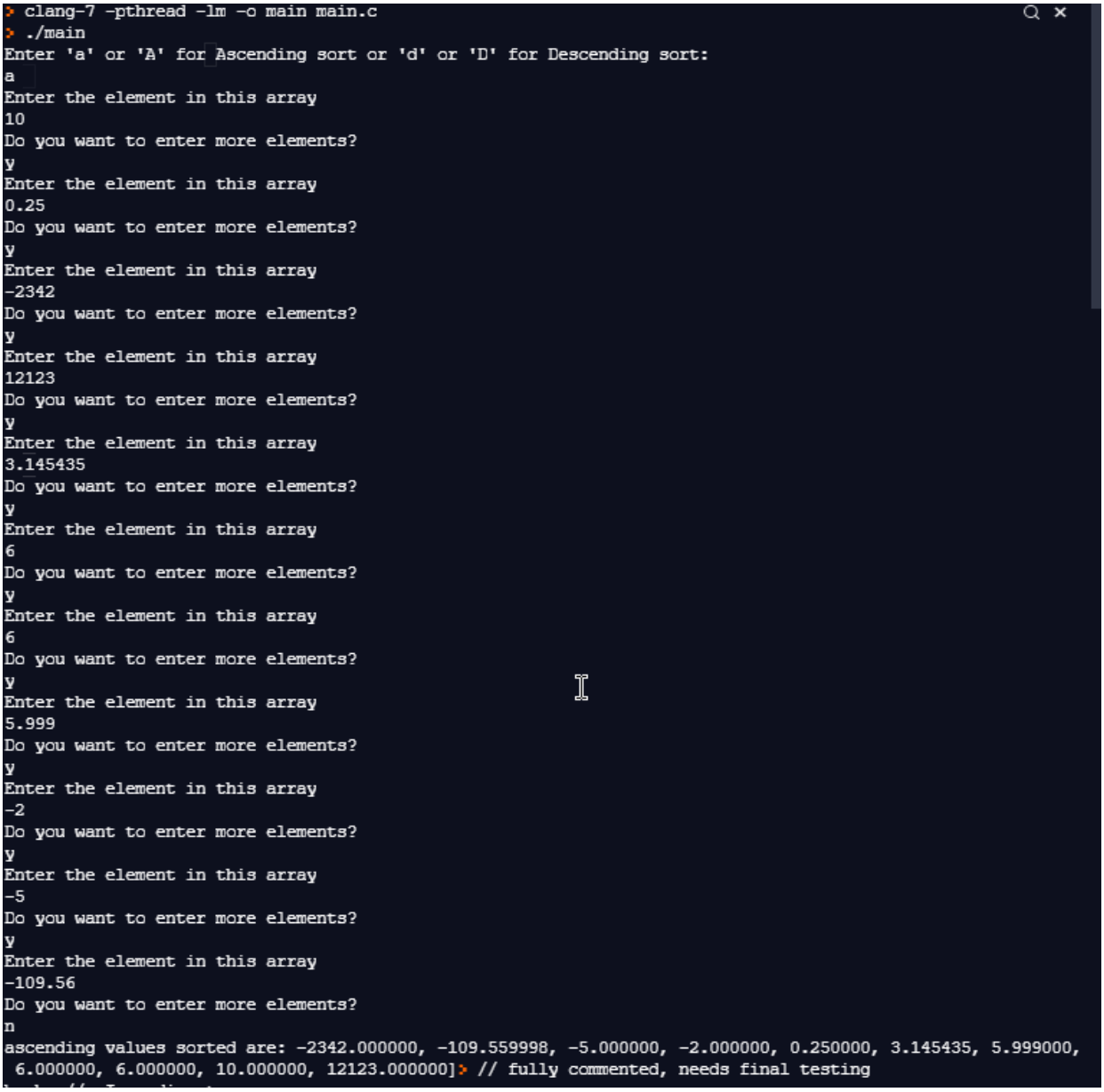
```
    // the size of the list will be allocated depending on the size provided by the user.



    for (int i = 0; i < val; i++) // depends on the size the user wishes, i'll just do the test case here.

    {

        printf("enter string: ");      // enter user input

        list[i] = (char *)malloc(sizeof(list)*10); /// 100 bc i've had issues with the sizing, dynamic allocation. // the size will be allocated depending on the user input

        scanf("%s", list[i]); // scan the inputs

    }

    sort_alphabetic(list);

    return 0;

}



void sort_alphabetic(char *the_list[])

{

    int i;        // will be used for iteration.
```

```
    int j;        // will be used for iteration.
    char *tmp;    // will be used for bubblesort
    char asc_des; // ask user if they want A or D
    printf("ascending or descending? ");
    scanf(" %c", &asc_des);


    for (i = 0; the_list[i]; i++) // double iteration for comparison
    {
        for (j = 0; the_list[j]; j++) // double iteration for comparison
        {

            if (asc_des == 'a' || asc_des == 'A')
            {
                if (strcmp(the_list[i], the_list[j]) < 0) // if strcmp between
two characters is less than 0, which means they are not the same and also
smaller in value
                {
                    tmp = the_list[i];         // bubblesort
```

```
                the_list[i] = the_list[j]; // bubblesort

                the_list[j] = tmp;         // bubblesort

            }

        }

        else if (asc_des == 'd' || asc_des == 'D')

        {

            if (strcmp(the_list[i], the_list[j]) > 0) // if it's larger,
which means it's z coming first

            {

                tmp = the_list[i];         // bubblesort

                the_list[i] = the_list[j]; // bubblesort

                the_list[j] = tmp;         // bubblesort

            }

        }

    }

}
```

```
    // personal addition here, i just like that the user knows which one they
picked.

    if (asc_des == 'a' || asc_des == 'A')

    {

        printf("in ascending order the list is \n");

    }

    else if (asc_des == 'd' || asc_des == 'D')

    {

        printf("in descending order the list is \n");

    }



    for (i = 0; the_list[i]; i++)

    {

        printf("%s\n", the_list[i]); // print the list

    }

}
```

QUESTION 3


> clang-7 -pthread -lm -o main main.c
> ./main
Enter 'a' or 'A' for Ascending sort or 'd' or 'D' for Descending sort:
a
Enter the element in this array
10
Do you want to enter more elements?
y
Enter the element in this array
0.25
Do you want to enter more elements?
y
Enter the element in this array
-2342
Do you want to enter more elements?
y
Enter the element in this array
12123
Do you want to enter more elements?
y
Enter the element in this array
3.145435
Do you want to enter more elements?
y
Enter the element in this array
6
Do you want to enter more elements?
y
Enter the element in this array
6
Do you want to enter more elements?
y
Enter the element in this array
5.999
Do you want to enter more elements?
y
Enter the element in this array
-2
Do you want to enter more elements?
y
Enter the element in this array
-5
Do you want to enter more elements?
y
Enter the element in this array
-109.56
Do you want to enter more elements?
n
ascending values sorted are: -2342.000000, -109.559998, -5.000000, -2.000000, 0.250000, 3.145435, 5.999000,
6.000000, 6.000000, 10.000000, 12123.000000]> // fully commented, needs final testing

HOW TO RUN IT:

gcc question3.c && ./a.out

Input whether you want ascending or descending first. Then, enter the values one by one to allocate space and let the program do its magic.

```c
#include <stdio.h>

#include <stdlib.h>

int sort_numeric(float *array, char input, int length); // the sort_numeric comes from q1.

int main()

{

int i, len = 0;

char inp, asc_des;

float *arr;                                 // pointer for storage.

arr = (float *)malloc(1 * sizeof(float)); // as requested, dynamic size allocation is done here. returns a void to pointer to float  here.

// c has automatic cast tho so idk if it's needed in c tbh i know cpp
```

```
        arr[0] = 0; // the values are added to this array



    printf("Enter 'a' or 'A' for Ascending sort or 'd' or 'D' for
                    Descending sort:\n");

                scanf("%s", &asc_des);

                    inp = 'y';



do // there's multiple ways this could be done, i prefer the do-while

                            {

            printf("Enter the element in this array\n");

                        scanf("%f", &arr[i]);

                            i = i + 1;

    printf("Do you want to enter more elements?\n"); // we need to
                ask the user each time

                        scanf("%s", &inp);

                    if (inp == 'y' || inp == 'Y')

                            {

                arr = realloc(arr, (i + 1) * sizeof(float));
```

```
                                    }

                } while (inp == 'y' || inp == 'Y');



            len += i; // iterate, then send to sort_numeric()

                    sort_numeric(arr, asc_des, len);

                                return 0;

                                }

        int sort_numeric(float *array, char input, int len)

                                {

                            int first, second, i;

    double tmp;                         // temp variable will be used for
                    bubble sort later on, check L17

        if (input == 'a' || input == 'A') // if the input is A or A

                                    {



            for (first = 0; first < len; first++) // until the length of
                        the array is finished

                                    {
```

```
                    for (second = first + 1; second < len; second++)

                                        {

                                if (*(array + first) >= *(array + second)) //
if array [the user input] plus first is larger than the second then using temp
                        switch their positions. Like a bubble sort.

                                            {



                                        tmp = *(array + first);
                                // compare-switch

                                    *(array + first) = *(array + second);
                                // compare-switch

                                    *(array + second) = tmp;
                                // compare-switch

                                            }

                                        }

                                    }

                                }

                    else if (input == 'd' || input == 'D')
```

```
                                        {

                                for (first = 0; first < len; first++)

                                        {

                                for (second = first + 1; second < len; second++)

                                        {

                                if (*(array + first) <= *(array + second)) //
if array [the user input] plus first is less than the second then using temp
                                switch their positions. like a bubble sort.

                                        {



                                        tmp = *(array + first);
                                // compare-switch

                                        *(array + first) = *(array + second);
                                //compare-switch

                                        *(array + second) = tmp;
                                // compare-switch

                                        }

                                    }

                                }
```

```
                }



                else

                {

        printf("option not available .\n"); // return an error if it's
                        malfunctioning

                return 0;

                }

    // just a little personal addition here, tell the user if their inputs
                are ascending descending or an error.

            if (input == 'a' || input == 'A')

                {

                printf("ascending");

                }

        else if (input == 'd' || input == 'D')

                {

                printf("descending");

                }
```

```
                else

                {

            printf("error");

                }


        printf(" values sorted are: ");


          for (i = 0; i < len; i++)

                {

            if (i != len - 1)

                    {

            printf("%f, ", array[i]);

                    }

                else

                    {

            printf("%f]", array[i]);

                    }
```

```
    }

return 0;

}
```

QUESTION 4

```
Enter the number of people you want to register:
10
Enter details for Person # 1:
first name: evonne
last name ostrem
Enter Birth Date(mm/dd/yyyy) : 07/21/1985
Choose sex :
    1. Male
    2. Female
    3. Do not wish to identify
     Enter choice : 2
Enter Dose Number : 1
Choose type of vaccine :
    1. Pfizer
    2. Moderna
    3. Johnson&Johnson
enter choice, 2
Enter Zip : 30022
Enter details for Person # 2:
first name: vanessa
last name kilman
Enter Birth Date(mm/dd/yyyy) : 10/11/2001
Choose sex :
    1. Male
    2. Female
    3. Do not wish to identify
     Enter choice : 2
Enter Dose Number : 1
Choose type of vaccine :
    1. Pfizer
    2. Moderna
    3. Johnson&Johnson
enter choice, 1
Enter Zip : 30009
Enter details for Person # 3:
first name: basilia
last name stephenson
Enter Birth Date(mm/dd/yyyy) : 05/23/1982
Choose sex :
    1. Male
    2. Female
    3. Do not wish to identify
     Enter choice : 3
Enter Dose Number : 1
Choose type of vaccine :
    1. Pfizer
    2. Moderna
    3. Johnson&Johnson
enter choice, 3
Enter Zip : 30021
```

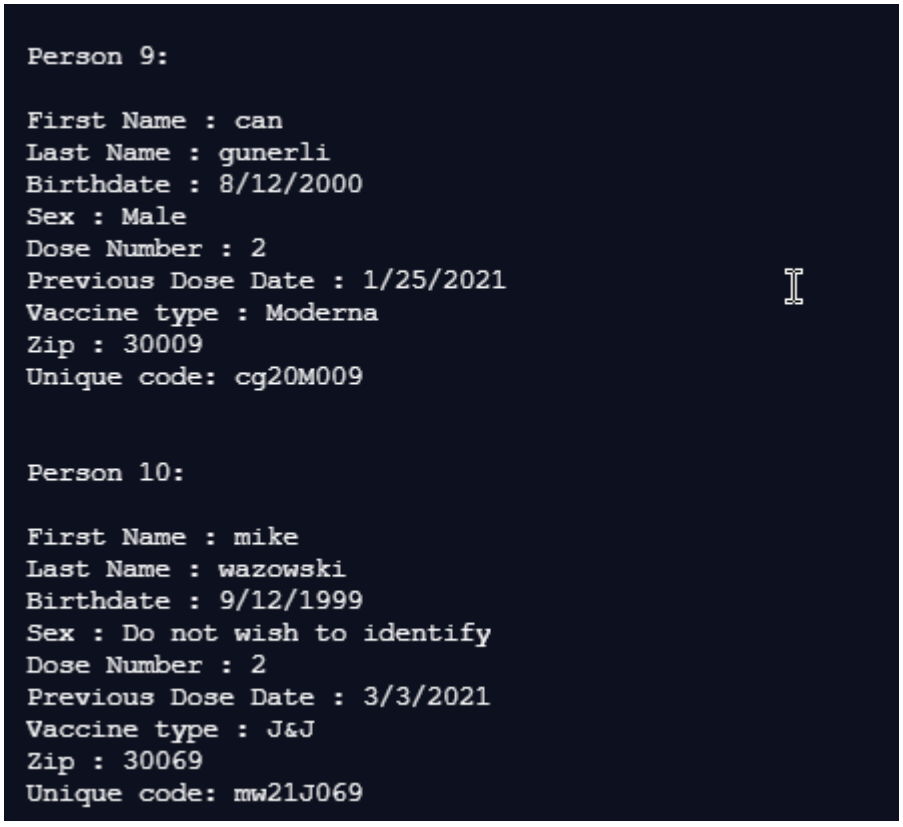
```
Enter details for Person # 4:
first name: raymundo
last name remigio
Enter Birth Date(mm/dd/yyyy) : 07/30/1982
Choose sex :
    1. Male
    2. Female
    3. Do not wish to identify
     Enter choice : 1
Enter Dose Number : 1
Choose type of vaccine :
    1. Pfizer
    2. Moderna
    3. Johnson&Johnson
enter choice, 3
Enter Zip : 30069
Enter details for Person # 5:
first name: veronica
last name juares
Enter Birth Date(mm/dd/yyyy) : 02/02/1984
Choose sex :
    1. Male
    2. Female
    3. Do not wish to identify
     Enter choice : 2
Enter Dose Number : 2
Enter Previous Dose Date(mm/dd/yyyy) : 04/15/2021
Choose type of vaccine :
    1. Pfizer
    2. Moderna
    3. Johnson&Johnson
enter choice, 3
Enter Zip : 16902
Enter details for Person # 6:
first name: chantay
last name cherry
Enter Birth Date(mm/dd/yyyy) : 01/01/2021
Choose sex :
    1. Male
    2. Female
    3. Do not wish to identify
     Enter choice : 3
Enter Dose Number : 2
Enter Previous Dose Date(mm/dd/yyyy) : 02/01/2021
Choose type of vaccine :
    1. Pfizer
    2. Moderna
    3. Johnson&Johnson
enter choice, 1
Enter Zip : 124578
```

```
Enter details for Person # 7:
first name: sammie
last name kneip
Enter Birth Date(mm/dd/yyyy) : 12/10/1984
Choose sex :
    1. Male
    2. Female
    3. Do not wish to identify
     Enter choice : 2
Enter Dose Number : 2
Enter Previous Dose Date(mm/dd/yyyy) : 03/01/2021
Choose type of vaccine :
    1. Pfizer
    2. Moderna
    3. Johnson&Johnson
enter choice, 2
Enter Zip : 123456
Enter details for Person # 8:
first name: barbera
last name gentry
Enter Birth Date(mm/dd/yyyy) : 12/05/2000
Choose sex :
    1. Male
    2. Female
    3. Do not wish to identify
     Enter choice : 2
Enter Dose Number : 2
Enter Previous Dose Date(mm/dd/yyyy) : 02/05/2021
Choose type of vaccine :
    1. Pfizer
    2. Moderna
    3. Johnson&Johnson
enter choice, 2
Enter Zip : 192390
Enter details for Person # 9:
first name: can
last name gunerli
Enter Birth Date(mm/dd/yyyy) : 08/12/2000
Choose sex :
    1. Male
    2. Female
    3. Do not wish to identify
     Enter choice : 1
Enter Dose Number : 2
Enter Previous Dose Date(mm/dd/yyyy) : 01/25/2021
Choose type of vaccine :
    1. Pfizer
    2. Moderna
    3. Johnson&Johnson
enter choice, 2
```

```
enter choice, 2
Enter Zip : 30009
Enter details for Person # 10:
first name: mike
last name wazowski
Enter Birth Date(mm/dd/yyyy) : 09/12/1999
Choose sex :
    1. Male
    2. Female
    3. Do not wish to identify
     Enter choice : 3
Enter Dose Number : 2
Enter Previous Dose Date(mm/dd/yyyy) : 03/03/2021
Choose type of vaccine :
    1. Pfizer
    2. Moderna
    3. Johnson&Johnson
enter choice, 3
Enter Zip : 30069
Information about registered people
```

```
Person 1:

First Name : evonne
Last Name : ostrem
Birthdate : 7/21/1985
Sex : Female
Dose Number : 1
Vaccine type : Moderna
Zip : 30022
Unique code: eo35M022


Person 2:

First Name : vanessa
Last Name : kilman
Birthdate : 10/11/2001
Sex : Female
Dose Number : 1
Vaccine type : Pfizer
Zip : 30009
Unique code: vk19P009


Person 3:

First Name : basilia
Last Name : stephenson
Birthdate : 5/23/1982
Sex : Do not wish to identify
Dose Number : 1
Vaccine type : J&J
Zip : 30021
Unique code: bs38J021


Person 4:

First Name : raymundo
Last Name : remigio
Birthdate : 7/30/1982
Sex : Male
Dose Number : 1
Vaccine type : J&J
Zip : 30069
Unique code: rr38J069
```

```
Person 5:

First Name : veronica
Last Name : juares
Birthdate : 2/2/1984
Sex : Female
Dose Number : 2
Previous Dose Date : 4/15/2021
Vaccine type : J&J
Zip : 16902
Unique code: vj37J902


Person 6:

First Name : chantay
Last Name : cherry
Birthdate : 1/1/2021
Sex : Do not wish to identify
Dose Number : 2
Previous Dose Date : 2/1/2021
Vaccine type : Pfizer
Zip : 124578
Unique code: cc00P457


Person 7:

First Name : sammie
Last Name : kneip
Birthdate : 12/10/1984
Sex : Female
Dose Number : 2
Previous Dose Date : 3/1/2021
Vaccine type : Moderna
Zip : 123456
Unique code: sk36M345


Person 8:

First Name : barbera
Last Name : gentry
Birthdate : 12/5/2000
Sex : Female
Dose Number : 2
Previous Dose Date : 2/5/2021
Vaccine type : Moderna
Zip : 192390
Unique code: bg20M239
```



HOW TO RUN IT:

gcc question4.c && ./a.out

Input how many people you want to keep the details of, for our example it is 10.

Input name, last name ,birthdate ,sex ,dose ,and other information accoridngly and you will be provided with the final outlook as well as a unique code generated for you.

```c
#include <stdio.h>

#include <string.h>

#include <time.h> // i'll use time to do agetwodates, which is a function to retrive the difference of years/days/months of the person's birthday, since that's needed for generate_code()

//structure to hold data related user



struct user

{ // first create a struct to simplify our job.

  char fname[100];

  char lname[100];

  int dd, pdd; // day

  int mm, pmm; // month

  int yy, pyy; // year

  char sex[100];

  int dnumber;

  char vaccinetype[100];
```

```
  char zip[100];

};

// they need to be declared here unless you want out-of-scope declaration errors and implicit declarations



void regis(int);

int agetwodates(int pdate, int pmonth, int pyear, int bdate, int bmonth, int byear);

void gen_code(struct user var);

void display(struct user var, int i);



// letting c know that i'll be calling these methods here. in total there're five functions/methods.



int main(void)

{



  int how_many_people; // this will be given by the user. for now, i'll test up to 10 as requested.
```

```c
  printf("Enter the number of people you want to register:\n");

  scanf("%d", &how_many_people);

  regis(how_many_people);

  return 0;

}

// in order to get age between two days, i'll use this function called agetwodates

int agetwodates(int pdate, int pmonth, int pyear, int bdate, int bmonth, int byear)

{

  int month[] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31}; // the months of the year feb is 28 so

  if (bdate > pdate)

  {

    pdate = pdate + month[bmonth - 1];

    pmonth = pmonth - 1;

  }

  if (bmonth > pmonth)
```

```
  {
    pyear = pyear - 1;
    pmonth = pmonth + 12;
  }
  int final_date = pdate - bdate;
  int final_month = pmonth - bmonth;
  int final_year = pyear - byear;
  return final_year;
}
void regis(int len)
{
  int choice;
  char id[9];
  char age[3];
  int amount_of_ppl;
  amount_of_ppl = len;
  struct user person[amount_of_ppl];
```

```
  int i;

  for (i = 0; i < amount_of_ppl; i++)

  {

    printf("Enter details for Person # %d:\n", i + 1);



    printf("first name: ");

    scanf("%s", person[i].fname);



    printf("last name ");

    scanf("%s", person[i].lname);



    printf("Enter Birth Date(mm/dd/yyyy) : ");                    // we
don't live in europe so, this would need to be changed if it's the uk

    scanf("%d/%d/%d", &person[i].mm, &person[i].dd, &person[i].yy); // put the
values in order to the struct



    printf("Choose sex : \n");

    printf("\t1. Male\n");
```

```
printf("\t2. Female\n\t");

printf("3. Do not wish to identify");

printf("\n\t Enter choice : ");

scanf("%d", &choice);



if (choice == 1)

{

  strcpy(person[i].sex, "Male");

}

else if (choice == 2)

{

  strcpy(person[i].sex, "Female");

}

else if (choice == 3)

{

  strcpy(person[i].sex, "Do not wish to identify");

}
```

```
// i'm not gonna return an else, i don't think it's necessary.



printf("Enter Dose Number : ");

scanf("%d", &person[i].dnumber);



if (person[i].dnumber == 2)

{

  printf("Enter Previous Dose Date(mm/dd/yyyy) : ");

  scanf("%d/%d/%d", &person[i].pmm, &person[i].pdd, &person[i].pyy);

}



printf("Choose type of vaccine : \n");

printf("\t1. Pfizer\n");

printf("\t2. Moderna\n");

printf("\t3. Johnson&Johnson\n");

printf("enter choice, ");

scanf("%d", &choice);
```

```
    if (choice == 1)

    {

      strcpy(person[i].vaccinetype, "Pfizer");

    }

    else if (choice == 2)

    {

      strcpy(person[i].vaccinetype, "Moderna");

    }

    else if (choice == 3)

    {

      strcpy(person[i].vaccinetype, "J&J");

    }

    // i'm not gonna return an else, i don't think it's necessary. just string
copy is good enough.



    printf("Enter Zip : "); // enter zip and store it

    scanf("%s", &person[i].zip);
```

```
  }

  printf("Information about registered people \n"); // print the users using
display()

  for (i = 0; i < amount_of_ppl; i++)

  {

    display(person[i], i);

  }

}

void gen_code(struct user var)

{

  char id[9]; // the size would be this big for the randomly generated uniq
code



  id[0] = var.fname[0]; // get the f name first character

  id[1] = var.lname[0]; // get the l name first character



  time_t t = time(NULL);

  struct tm tm = *localtime(&t); // using time.h, get date
```

```
  int ageTemp = agetwodates(tm.tm_mday, tm.tm_mon + 1, tm.tm_year + 1900,
var.dd, var.mm, var.yy); // pass parameteres to get the person's bd



  id[2] = (char)(ageTemp / 10 + '0');

  id[3] = (char)(ageTemp % 10 + '0'); // age as the id's 4th element



  id[4] = var.vaccinetype[0];



  id[5] = var.zip[2]; // the first last three of the zipcode

  id[6] = var.zip[3];

  id[7] = var.zip[4];

  id[8] = '\0';

  printf("Unique code: %s\n\n", id); // unique code is created.

}

void display(struct user var, int i)

{
```

```
  printf("\nPerson %d:\n", i + 1);

  printf("\nFirst Name : %s\n", var.fname); // print results

  printf("Last Name : %s\n", var.lname);

  printf("Birthdate : %d/%d/%d\n", var.mm, var.dd, var.yy); // print in order.

  printf("Sex : %s\n", var.sex);

  printf("Dose Number : %d\n", var.dnumber);

  if (var.dnumber == 2)

  {

    printf("Previous Dose Date : %d/%d/%d\n", var.pmm, var.pdd, var.pyy); //
if the dose number is two, print that date as well.

  }

  printf("Vaccine type : %s\n", var.vaccinetype);

  printf("Zip : %s\n", var.zip);

  gen_code(var);

}
```