PROGRAMMING TECHNIQUES – 22/06/2023

The hard deadline to submit the self-evaluation report is 24/06/2023, at 23:59

SURNAME	
NAME	
MATRICOLA ID	

THEORY SECTION

EXERCISE 1 (5 points). - TO BE FILLED IN THE GIVEN PAPER!!!

Given the following sequence of pairs, where relation i-j indicates that vertex i is adjacent to vertex j:

13-6, 5-2, 1-9, 9-3, 12-9, 6-9, 4-8, 1-8

apply an on-line connectivity algorithm with quickunion. Nodes are named with integers between 0 and 13.

- What is the worst-time asymptotic complexity of the union and of the find operations with quickunion?

find: O(N) union: O(1)=Θ(1)

Justify your answer.

find requires walking a “chain” of nodes, whose length is at most the number of nodes N. Thus find is O(N).
union makes the representative of a “chain” of nodes point to the representative of another “chain” of nodes. This entails an assignment of a variable whose complexity is O(1)=Θ(1)

- In the Online connectivity algorithm is an explicit representation of the graph needed? Justify your answer.

No, an explicit representation of the graph is not needed. The data structure (array id) allows to keep track of what node represents (directly or indirectly) what node.

- What is the size of array id? Justify your answer.

Size is 14, as there are 14 integers in the range from 0 to 13. The size in bytes was not requested.

- Display array id as a sequence of integers after the step for pair 1-8

id: 0 9 2 8 8 2 3 7 8 3 10 11 3 6

- What represents node 6 at the end of the algorithm?

Node 8

EXERCISE 2 (5 points) TO BE FILLED IN THE GIVEN PAPER!!!

Sort the following array A of integers on 3 digits in base 10 in ascending order using radix sort:

200 643 387 820 272 607 743 294

- Under what condition is radix sort stable?

The algorithm used to sort columns one by one must be stable. Using counting sort satisfies this condition.

- step 1 unit column: show the initial content of array C of simple and multiple occurrences as a sequence of integers

simple occurrences C = 2, 0, 1, 2, 1, 0, 0, 2, 0, 0

multiple occurrences C = 2, 2, 3, 5, 6, 6, 6, 8, 8, 8

- step 2 column of tens: show as a sequence of integers the content of array A after the counting sort step

A = 200, 607, 820, 643, 743, 272, 387, 294

- what is the complexity of radix sort for sorting N d-digit integers (in base 10)?

X O(dN)
□ O(N)
□ O(NlogN)
□ O(dN^2)

Briefly justify your answer:

worst-case asymptotic complexity of Counting sort is T(N) = O(N+k), where items to sort are integers in the range (0… k-1). Counting sort runs d times. Complexity is T(N) = O(d(N+k)). For numbers in base 10, k is fixed and is 10, thus T(N) = O(dN)

- Assume you use insertion sort instead of counting sort to sort columns. What would be the complexity of radix sort for sorting N d-digit integers (in base 10)?

Answer: T(N) = O(dN^2)

Briefly justify your answer:

Worst-case asymptotic complexity of Insertion sort is T(N) = O(N^2). Insertion sort runs d times. Complexity is T(N) = O(dN^2).

NAME	
SURNAME	
MATRICOLA ID	

EXERCISE 3 (5 points) TO BE FILLED IN THE GIVEN PAPER!!!

Given the following piece of code:

```
typedef struct { int a; } structC;

typedef struct {
    int b;
    structC *next;
} structB;

typedef struct
{
    int c;
    structB *next;
} structA;

structA data1, *datap;
structB data2;
structC data3;
data1.c = 60;
data2.b = 30;
data3.a = 10;
datap = &data1;
data1.next = &data2;
data2.next = &data3;
```

1) For each of the following expressions, say if it is correct and, if so, what value is being accessed.

EXPRESSION	CORRECT (Y/N)	VALUE
(*datap).c	Y	60
datap->c	Y	60
datap.c	N	
data1.next->b	Y	30
datap->next->b	Y	30
datap.next.b	N	
datap->next.b	N	
(*(datap->next)).b	Y	30

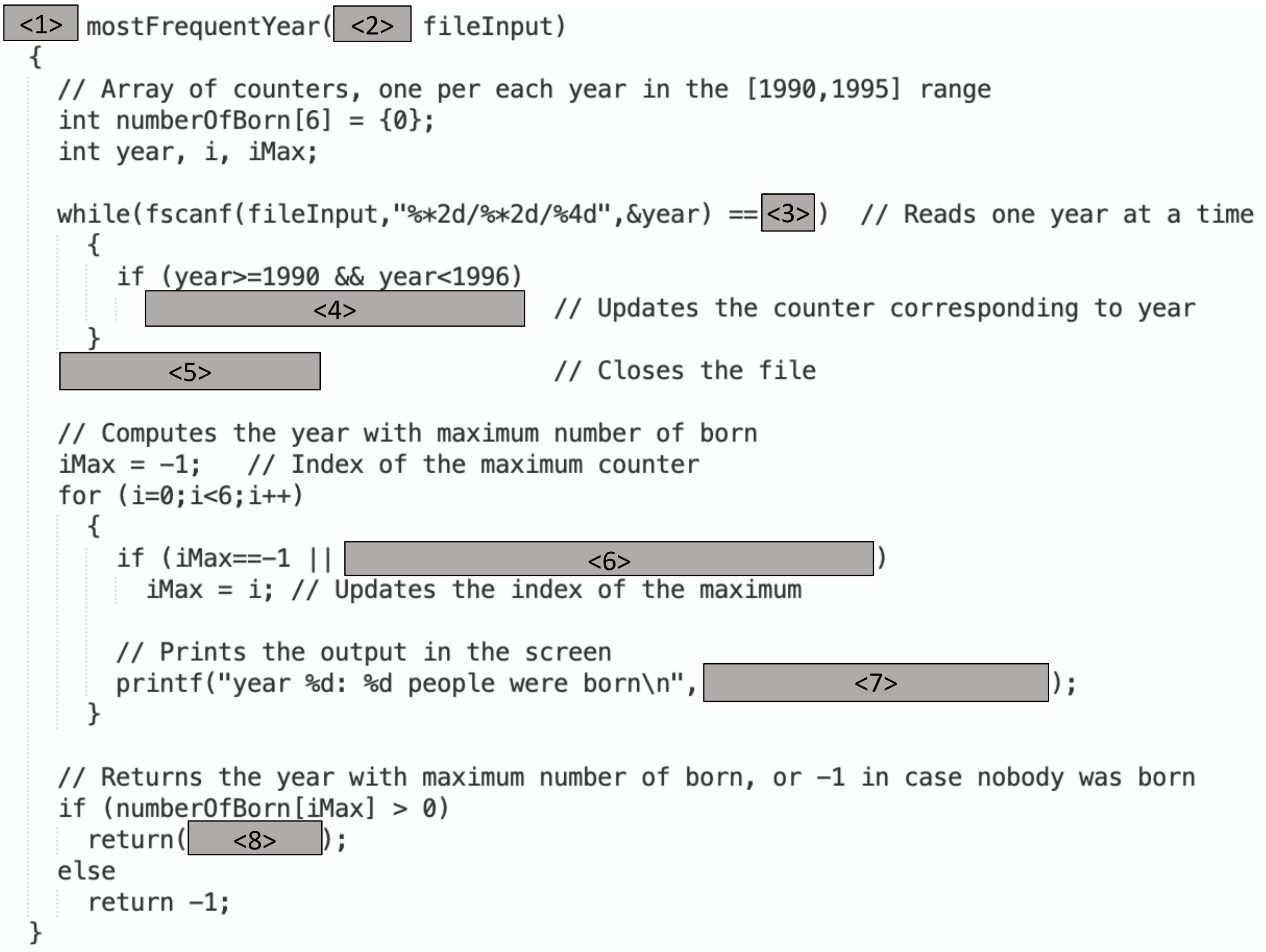
2) Which of the two variables `data2` and `datap` occupies more space in memory? Why?

`data2` occupies more space, as it has an integer as well as a pointer fields. Instead, `datap` is just a pointer. Then, `data2` occupies the same space as `datap,` plus the size of an integer.

PROGRAMMING SECTION (min threshold: 6/10 points on ex 4,5)

EXERCISE 4 (4 points)   TO BE FILLED IN THE GIVEN PAPER!!!

The function `mostFrequentYear` receives a pointer to a text file, already open, that contains a list of dates-of-birth of a group of people, in the dd/mm/yyyy format, one per line. The function computes and prints on the screen the number of births in each year between 1990 and 1995 (extremes included). Then, it returns to the caller the earliest year between 1990 and 1995 that had the maximum number of births, or -1 in case nobody was born in those years. Complete the given function, by filling the table below with the code that is missing in each of the corresponding boxes.



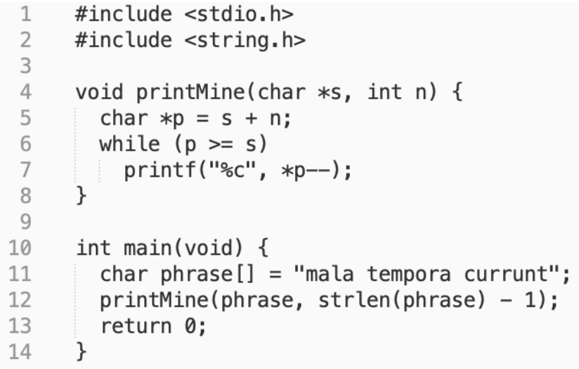
```
<1> mostFrequentYear( <2>  fileInput)
{
  // Array of counters, one per each year in the [1990,1995] range
  int numberOfBorn[6] = {0};
  int year, i, iMax;

  while(fscanf(fileInput,"%*2d/%*2d/%4d",&year) ==<3>)  // Reads one year at a time
    {
      if (year>=1990 && year<1996)
        <4>                                      // Updates the counter corresponding to year
    }
  <5>                                            // Closes the file

  // Computes the year with maximum number of born
  iMax = -1;   // Index of the maximum counter
  for (i=0;i<6;i++)
    {
      if (iMax==-1 || <6> )
        iMax = i; // Updates the index of the maximum

      // Prints the output in the screen
      printf("year %d: %d people were born\n", <7> );
    }

  // Returns the year with maximum number of born, or -1 in case nobody was born
  if (numberOfBorn[iMax] > 0)
    return( <8> );
  else
    return -1;
}
```

BLOCK	CODE
<1>	int
<2>	FILE*
<3>	1
<4>	numberOfBorn[year-1990]++;
<5>	fclose(fileInput);
<6>	numberOfBorn[i] > numberOfBorn[iMax]
<7>	i+1990, numberOfBorn[i]
<8>	iMax+1990

SURNAME	
NAME	
MATRICOLA ID	

EXERCISE 5 (6 points) TO BE FILLED IN THE GIVEN PAPER!!!

Given the following code, answer the questions below:

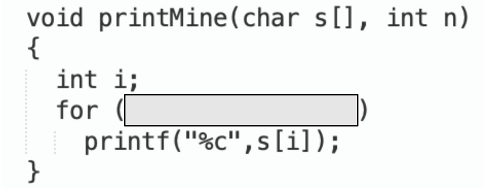


```
#include <stdio.h>
#include <string.h>

void printMine(char *s, int n) {
  char *p = s + n;
  while (p >= s)
    printf("%c", *p--);
}

int main(void) {
  char phrase[] = "mala tempora currunt";
  printMine(phrase, strlen(phrase) - 1);
  return 0;
}
```

1) What is the output of the given program?

tnurruc aropmet alam

2) See the following alternate implementation of the function `printMine`:

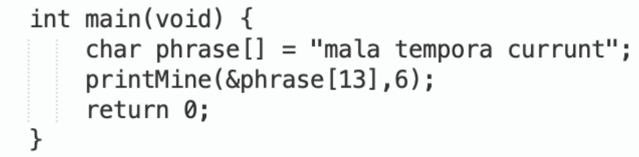


```
void printMine(char s[], int n)
{
  int i;
  for (                    )
    printf("%c",s[i]);
}
```

What should the code in the grey box be, for the function to behave the same as in the original code?

i=n; i>=0; i--

3) Suppose that the main is modified as follows:

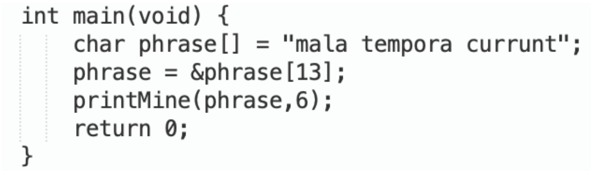


```
int main(void) {
    char phrase[] = "mala tempora currunt";
    printMine(&phrase[13],6);
    return 0;
}
```

Is it correct, syntactically? If no, explain why. If yes, what is the output of the program?

It is correct. The output is: tnurruc

4) Suppose that the main is modified as follows:



```
int main(void) {
    char phrase[] = "mala tempora currunt";
    phrase = &phrase[13];
    printMine(phrase,6);
    return 0;
}
```

Is it correct, syntactically? If no, explain why. If yes, what is the output of the program?

It is not correct. The array phrase cannot be reassigned as if it was a pointer variable.

EXERCISE 6 (8 points) USE YOUR OWN PAPER!!!

In a C program, a student is represented by the following struct:

```
typedef struct {
   char name[30];
   int n; // number of passed exams
   float avg; // average score of the passed exams
} student;
```

The data of a group of students (maximum 100) is reported in a text file, one student per line, in the following format:

```
name  score_1  score_2  score_3  … score_n
```

where each score is an integer value in the [18,30] range. Each student may have passed from 0 upto a maximum of 20 exams. Assume that names do not contain any spaces and have a maximum of 29 characters. There are no students with the same name.

1) Write a function `readStudents` that takes as input the name of the input file and stores the data of the students into an array of type `student`. The total number of students that were correctly read should be returned to the caller as return value. The function should have the following prototype:

```
int readStudents(char* filename, student s[], int n);
```

where `s` is the array of students where to store the data, and `n` the dimension of the array.

2) Write another function `sortStudents` that takes as input the array of students as they were read from the file, and sorts them by increasing average score, by implementing a Selection Sort as the sorting algorithm. The prototype of the function should be the following:

```
void sortStudents(student s[], int ns);
```

where `s` is the array of students and `ns` the number of students.

`NB.` You are required to implement only the given functions, and not the whole C program.

EXAMPLE OF INPUT FILE CONTENT:

```
Cicciopasticcio 18 23 30 21 18

Voldemort

Superwoman 30 28 30 30 30 29 30 30 30 30
```

PROPOSED SOLUTIONS

```
int readStudents(char *filename,student s[],int n)
{
  int score;
  FILE* fp = fopen(filename,"r");
  int ns=0;
  while(ns<n && fscanf(fp,"%s",s[ns].name) !=EOF )
    {
      s[ns].avg = 0.0;
      s[ns].n = 0;
      while(fscanf(fp,"%d",&score)==1)
      {
        s[ns].avg += score;
        s[ns].n++;
      }
      if (s[ns].n > 0)
        s[ns].avg/= s[ns].n;
      ns++;
    }
  return ns;
}


void sortStudents(student s[], int n){
  int i,j,iMin;
  student tmp;
  for (i=0;i<n-1;i++)
    {
      iMin = i;
      for (j=i+1;j<n;j++)

        {

          if (s[j].avg<s[iMin].avg)

            iMin = j;

        }

      tmp = s[i];

      s[i] = s[iMin];

      s[iMin] = tmp;

    }

}
```