

Computer Programming
3rd Presentation; Variant A
24-03-2023

1. (10 points) A term is a variable (letter) or a function (letter) of terms (at least one) within parantheses and separated by commas. Example: $f(g(x,y),h(z),t)$ Write a function that reads a term (ended by whitespace or EOF) and reports errors.
2. (10 points) Write a function that takes two unsigned, n and k and returns the number obtained by rotating the bits of n left by k positions (bits removed on the left are inserted at right). Handle arbitrary values for k (i.e. k might be above the number of bits of n).
3. (10 points) Implement a C function that will find, search and replace text strings. Receives three strings ($s1$, $s2$ and $s3$) and replaces all the occurrences of string $s2$ into the first string ($s1$), with the third string ($s3$), returning the new string. The replacement phrase can be shorter or longer than the searched for phrase. Show the updated initial line of text on the display.
4. (10 points) Implement a C program that checks whether an HTML file with path read from standard input (path less than 100 caracters) has all tables correct (all rows in a table have equal column counts). A table is enclosed between tags `<table>` and `</table>`. A table row is enclosed between tags `<tr>` and `</tr>` and a table cell between `</td>` and `</td>`. Whitespace between tags does not matter.

Notes

- Stick to the problem statement, asking for clarifications when in doubt. It is not enough to write some "compiling code", it had to do what is required.
- You are encouraged to define and implement any number of supplementary functions which you deem necessary for proving a good design of the solution
- Each problem shall be solved in a distinct C file representing a full working program: you will have to implement the `main()` function
- Name your files like: `NameSurname_VariantNo.c` where **Name** and **Surname** are replaced accordingly, **Variant** is taken from the boxed header and **No** is the number of the item you are solving.
- Solutions provided by Generative Pre-trained Transformers are not allowed!

Computer Programming
3rd Presentation; Variant B
24-03-2023

1. (10 points) A prefix expression is an unsigned number or, in parantheses (), an operator `+` `*` followed by ≥ 1 whitespace-separated expressions to be added or multiplied. Example: $(+ (* 2 3 4) 5 6)$ is 35. Write a function that reads an expression and returns its value.
2. (10 points) Write a function which "encodes" a pair of uint 32 s given as arguments into a single uint 64 it returns. The returned value binary representation is composed by interleaving the binary representations of the 2 original arguments, thus merging a pair of "bits" from the 2 original numbers
3. (10 points) Write a function to read a text over several lines until end of file and then print only those lines containing a "word" (string made of only non-whitespace characters), received as parameter. **Note** that you have to write your own string processing functions (you should NOT use functions from `string.h`)
4. (10 points) Write a function that reads a file named on the command line and prints all words in increasing length order, breaking ties alphabetically. Words are strings of non-whitespace characters. You may truncate any words longer than 63 chars. The file is arbitrarily large. Be efficient!

Notes

- Stick to the problem statement, asking for clarifications when in doubt. It is not enough to write some "compiling code", it had to do what is required.
- You are encouraged to define and implement any number of supplementary functions which you deem necessary for proving a good design of the solution
- Each problem shall be solved in a distinct C file representing a full working program: you will have to implement the `main()` function
- Name your files like: `NameSurname_VariantNo.c` where **Name** and **Surname** are replaced accordingly, **Variant** is taken from the boxed header and **No** is the number of the item you are solving.
- Solutions provided by Generative Pre-trained Transformers are not allowed!

Computer Programming
2nd Presentation; Variant B
24-02-2023

1. (10 points) Implement a function that reads characters from standard input, consisting of integers, separated by a comma, and returns the length (number of integers) of the longest sub-sequence consisting of only prime numbers. Handle common errors by returning -1.
2. (10 points) Implement a function that receives two, exactly 64-bit, unsigned values and returns an 8-bit unsigned number having bit on position **i** set if-and-only-if the bytes in position **i** of both the arguments are having the same value.
Note: You shall use C bitwise operators
3. (3 points) Write a function `double* copyFloats(const char *s, unsigned *n);` that returns an array of doubles, consisting of all the float constants found into string *s*. Consider floats as being written in any acceptable C format. Set value pointed by *n* to the number of elements from the returned array. You shall NOT use `strtok!!!`
4. (10 points) Write a function that prints in alphabetical order all headings from an HTML file named on the command line. A heading is enclosed between tags `<hN>` and `</hN>` with *N* a digit from 1 to 6. Headings are at most 100 chars long, but arbitrarily many.
I.e.: a possible input is `<h2>Upt</h2> <h3>Acsa</h3>` which would print `Acsa Upt`

Notes

- Stick to the problem statement, asking for clarifications when in doubt. It is not enough to write some "compiling code", it had to do what is required.
- You are encouraged to define and implement any number of supplementary functions which you deem necessary for proving a good design of the solution
- Each problem shall be solved in a distinct C file representing a full working program: you will have to implement the `main()` function
- Name your files like: `NameSurname_VariantNo.c` where **Name** and **Surname** are replaced accordingly, **Variant** is taken from the boxed header and **No** is the number of the item you are solving.
- Solutions provided by Generative Pre-trained Transformers are not allowed!

Computer Programming
2nd Presentation; Variant B
24-02-2023

1. (2 points) A string can be changed in two ways: either append its last character, or append its length modulo 10, as a digit. Write a function `void printAllStrings(char *s1, unsigned char k);` that takes a positive *k* and prints all strings that can be obtained from strings denoted by *s1* after making exactly *k* changes.
2. (10 points) Implement a function which receives an `uint16_t a` number and another `uint64_t b` and checks whether all the nibbles of *a* are to be found into *b* (in the same order but not necessarily adjacent)
I.e. `a=0xAB12` and `b=0xAD2B32DFF1C278` is a pair of such numbers;
3. (10 points) Implement a function that receives C-styled string and returns a copy of the string where all the digits are replaced with their English spelling. I.e. `f("Ana are 12 mere si 13.5 pere")` should return `"Ana are onetwo mere si onethree.five pere"`
4. (10 points) Write a program that prints in order of the numerical values all the registration plates belonging to cars registered in a particular county, from a list found in a file with it's path read from standard input.
Consider the common Romanian registration plate of the format: **County-Number-Id** (I.e. `TM-01-ABC`) (the Number part might be represented on two or three digits). Break ties by the lexicographical values of the **Id**. Normalize all the values to uppercase. Skip all the records which don't match the pattern.

Notes

- Stick to the problem statement, asking for clarifications when in doubt. It is not enough to write some "compiling code", it had to do what is required.
- You are encouraged to define and implement any number of supplementary functions which you deem necessary for proving a good design of the solution
- Each problem shall be solved in a distinct C file representing a full working program: you will have to implement the `main()` function
- Name your files like: `NameSurname_VariantNo.c` where **Name** and **Surname** are replaced accordingly, **Variant** is taken from the boxed header and **No** is the number of the item you are solving.
- Solutions provided by Generative Pre-trained Transformers are not allowed!

Computer Programming
1st Presentation; Variant A
26-01-2023

1. (10 points) An expression is a sum (+) of one or more products (*) of one or more unsigned numbers, with optional whitespace. Write a function that reads an expression from input and computes its value.
2. (10 points) Implement a C function that receives a 64-bit unsigned number and returns the number made of the last digit of the lengths of contiguous sub-sequences of identical bits in the argument
3. (10 points) Implement a function that receives a C-string, and a list of separator characters and returns a NULL-terminated array of copies to the "words" found in the input string. We consider as "word" a sequence of characters delimited by any of the characters from the list of separators. You shall NOT use strtok!!!
4. (10 points) Implement a function that receives as argument a C-string representing a file path to a Unix password file (with the structure described below) and prints the list of unix-groups in descending number of users assigned to those groups. The Unix password file structure is as follows: **Username:X:UserId:GroupId:EXTRA_FIELDS** where you are to use the GroupId to identify the group to which each UserName is assigned. **You might assume a program-level defined maximum number of rows in the file**

Notes

- Stick to the problem statement, asking for clarifications when in doubt. It is not enough to write some "compiling code", it had to do what is required.
- You are encouraged to define and implement any number of supplementary functions which you deem necessary for proving a good design of the solution
- Each problem shall be solved in a distinct C file representing a full working program: you will have to implement the `main()` function
- Name your files like: **NameSurname_VariantNo.c** where **Name** and **Surname** are replaced accordingly, **Variant** is taken from the boxed header and **No** is the number of the item you are solving.
- Solutions provided by Generative Pre-trained Transformers are not allowed!

Computer Programming
1st Presentation; Variant B
26-01-2023

1. (10 points) An expression is a sequence of terms. A term is an unsigned number or a sum (+) of unsigned numbers enclosed in (). Whitespace is optional. Write a function that reads an expression from input and computes its value, defined as the product of the terms.
2. (10 points) Implement a function that receives a natural number and returns a 64-bit unsigned value so that, from least significant digit/bit to most significant digit/bit there is a number of bits of 1 equal to the number denoted by the even digits in the argument and a number of bits of 0 equal to the number denoted by odd digits in the argument
3. (10 points) Implement a function that receives an array of pointers to C-strings and returns a C-string obtained by concatenating the strings from the array which are actually C-floating point numbers (in any C format)
4. (10 points) Implement a function that receives as argument a C-string representing a file path to a Unix system log file (with the structure described below) and prints the days recorded in the log, in descending order of the number of logged events. The Unix system log has the following structure:
Month Day Hour:Min:sec hostname A.String_Representing_The_Message_until_end_of_line
To the right of *hostname* there is an arbitrary text, until end-of-line which you shall ignore for the purpose of this assignment.

Notes

- Stick to the problem statement, asking for clarifications when in doubt. It is not enough to write some "compiling code", it had to do what is required.
- You are encouraged to define and implement any number of supplementary functions which you deem necessary for proving a good design of the solution
- Each problem shall be solved in a distinct C file representing a full working program: you will have to implement the `main()` function
- Name your files like: **NameSurname_VariantNo.c** where **Name** and **Surname** are replaced accordingly, **Variant** is taken from the boxed header and **No** is the number of the item you are solving.
- Solutions provided by Generative Pre-trained Transformers are not allowed!

Computer Programming Intro Test 06-10-2023
--

Solve any 4 problems of the following:

1. (10 points) Implement a function that reads from standard input data organized as CSV (comma-separated values) and for each line prints how many of the "fields consist only of alphanumerical characters (as given by `isalnum()`). Note: input is arbitrary long.
2. (10 points) Write a function returning the maximum of all values obtained by repeated application of `swaprot()` to a 32-bit integer value received as a parameter. `swaprot()` computes a value as follows: swaps the two 16-bit halves and rotates right by 4 bits the obtained number (rightmost bits are inserted leftmost).
3. (10 points) Implement a function that receives a C-styled string and also an array of doubles, together with its capacity, and copies as many floating-point values from the string (written in any C format) into the array, without exceeding the capacity and returns the number of copied values.
4. (10 points) Write a program that reads from a file named on the command line, containing lines of the form `identifier=string-typed-value`. Print the lines sorted lexicographically by the "value". Lines have at most 80 chars. Identifiers are like in C. Ignore incorrect lines.
5. (10 points) An expression is a sequence of terms. A term is an unsigned number or a sum (+) of unsigned numbers enclosed in (). Whitespace is optional. Write a function that reads an expression from input and computes its value, defined as the product of the terms.

Suggestions:

- All solutions should be compilable under gcc using the C language (std99 is allowed);
- Only safe warnings are allowed, but you should strive to eliminate even those;
- Each problem shall be solved in a single, distinct file;
- Keep to the problem statement, asking for clarifications when in doubt.
- Put all your skills in the solution you are handling: naming conventions, code indenting, safe code and so on.
- You want to prove you have good programming skills
- You might provide a convoluted solution to a problem, as long as it sticks to the problem statement and proves you master the basic concepts well enough, but always should strive for an elegant one

Computer Programming
3rd. Attempt; Part 1
01-04-2022

1. (10 points) Implement a function which reads characters from standard input, consisting of integers, separated by comma and returns the length (number of integers) of the longest sub-sequence consisting of only prime numbers. Handle common errors by returning -1.
2. (10 points) Implement a function that receives two, exactly 64-bit, unsigned values and returns an 8-bit unsigned number having bit on position **i** set if-and-only-if the bytes in position **i** of both the arguments are having the same value.
Note: You shall use C bitwise operators

Notes

- Stick to the problem statement, asking for clarifications when in doubt.
- Each problem shall be solved in a distinct C file representing a full working program: you will have to implement the `main()` function
- Name your files like: `NameSurname_1.c` where **Name** and **Surname** are replaced accordingly, and the number is the number of the problem you are solving.

Computer Programming
3rd. Attempt; Part 2
01-04-2022

1. (10 points) Implement a function that receives C-styled string and returns a copy of the string where all the digits are replaced with their English spelling. **I.e.** `f("Ana are 12 mere si 13.5 pere")` should return `"Ana are onetwo mere si onethree.five pere"`
2. (10 points) Write a program that prints in numerical value all the registration plates belonging to cars registred in a particular county, from a list in file named on command line.
Consider the common Romanian registration plate of the format: **County-Number-Id** (I.e. **TM-01-ABC**) (the Number part might be represented on two or three digits). Break ties by the lexicographical values of the **Id**. Normalize all the values to uppercase.

Notes

- Stick to the problem statement, asking for clarifications when in doubt.
- Each problem shall be solved in a distinct C file representing a full working program: you will have to implement the `main()` function
- Name your files like: `NameSurname_1.c` where **Name** and **Surname** are replaced accordingly, and the number is the number of the problem you are solving.

1. (10 points) Se considera un fisier text care pe prima linie are un numar natural n , pe a doua linie o lista de n numere intregi iar pe urmatoarele linii (pana la sfarsitul fisierului) perechi de numere intregi a, b si determina intervalele $[a, b]$ care includ cel putin un element din lista de valori aflata pe randul al doilea al fisierului

Implementati un program C care primeste ca si parametru o cale de fisier text cu formatul descris mai sus si afiseaza pe iesirea standard lista intervalelor care satisfac conditia prezentata, incheiata cu numaru de asemenea intervale. Algoritmul va trebui sa aiba complexitate subliniara.

2. (10 points) Implementati o biblioteca, in sensul limbajului C, care permite lucrul cu esantioane de temperatura (tipul `Esantion_t`), colectate de-a lungul unei perioade de timp. Fiecare esantion pastreaza informatii despre *valoarea masurata*, *momentul de timp al masurarii*, *locatia masurarii* (coordonate GPS in format grade, minute si secunde) si posibile *comentarii*. Biblioteca va oferi urmatoarele operatii publice:

```
Esantion_t makeEsantion(/*puneti ce parametri considerati*/);
```

```
/* functia construiesc un element de tip Esantion_t pe baza valorilor componente */
```

```
Esantion_t* addSample(Esantion_t* t, int* n, Esantion_t e);
```

```
/*functia adauga un esantion la vectorul de esantioane t, avand la momentul de fata n elemente  
dupa adaugare se actualizeaza adecvat valoarea lui n*/
```

```
Esantion_t getMaxTemp(/*puneti ce parametri considerati*/);
```

```
/* returneaza Esantionul care are valoarea maxima pentru temperatura */
```

```
Esantion_t* copySamples(Esantion_t* v[], int n, double tMin, double tMax);
```

```
/*functia creeaza si returneaza un nou tablou in care copiaza Esantioanele din v care au  
temperatura cuprinsa intre cele doua limite (inclusive).
```

```
Tabloul se incheie cu un element care are anul masurarii 0*/
```

Implementati un program-client in care sa testati si ilustrati functionalitatile.

1. (10 points) Se considera un fisier care contine un sir de numere naturale din intervalul $[0, 100]$, separate prin câte un spațiu. Se cere să se determine toate perechile distincte formate din termeni ai șirului aflat în fișier, x și y ($y - x \geq 2$), astfel încât să nu existe niciun termen al șirului care să aparțină intervalului (x, y) . Numerele din fiecare pereche sunt afișate pe câte o linie a ecranului, în ordine strict crescătoare, separate printr-un spațiu, iar dacă nu există nicio astfel de pereche, se afișează pe ecran mesajul **nu exista**. Implementati un program C care primeste calea catre fisier ca si parametru in linie de comanda si afiseaza rezultatul la iesirea standard. Se va oferi un algoritm cu complexitate in timp maxim liniara.

2. (10 points) Implementati o biblioteca, in sensul limbajului C, care permite lucrul cu Pixeli (tipul `Pixel_t`). Fiecare pixel pastreaza informatii despre *coordonatele X si Y*, *culoarea sa* sub forma *tripletei R, G, B*, *transparenta sa* sub forma *intensitatii alpha* (valoare reala intre 0-perfect transparent si 1-perfect opac) Biblioteca va oferi urmatoarele operatii publice:

```
Pixel_t makePixel(/*puneti ce parametri considerati*/);
```

```
/*construiesc si reține un Pixel cu informatiile necesare */
```

```
Pixel_t* addPixel(Pixel_t v[], int n, Pixel_t p);
```

```
/*functia adauga un Pixel la vectorul de Pixeli v, avand la momentul de fata n elemente;  
dupa adaugare se actualizeaza adecvat valoarea lui n*/
```

```
Esantion_t getMostOpaquePixel(/*puneti ce parametri considerati*/);
```

```
/* returneaza pixelul cu valoarea cea mai mare a lui alpha; daca exista mai multe asemenea va  
Pixel_t* copyPixels(Pixel_t* v[], int n, double dMin, double dMax);
```

```
/*functia creeaza si returneaza un nou tablou in care copiaza Pixelii din v care au  
distranta pana la origine (coordonate (0,0)) cuprinsa intre dMin si dMax.
```

```
Tabloul se incheie cu un element care are coordonatele (-1, -1).*/
```

Implementati un program-client in care sa testati si ilustrati functionalitatile.

Computer Programming Initial test 11-10-2019
--

1. (2 points) A string can be changed in two ways: either append its last character, or append its length modulo 10, as a digit. Write a function `void printAllStrings(char *s1, unsigned char k);` that takes a positive k and prints all strings that can be obtained from strings denoted by `s1` after making exactly k changes.
2. (2 points) Implement a function `unsigned long getMaxRightShiftRotate(unsigned long x);` which returns the maximum number which can be obtained from x by shifting nibbles of x to the right, with rotation.
3. (3 points) Write a function `double* copyFloats(const char *s, unsigned *n);` that returns an array of doubles, consisting of all the float constants found into string s . Consider floats as being written in any acceptable C format. Set value pointed by n to the number of elements from the returned array. processed
4. (3 points) Write a program that reads a file named on the command line, containing lines of the form `identifier=any-char-string`. Print the lines sorted alphabetically by identifier. Lines have at most 80 chars. Identifiers are like in C. Ignore incorrect lines.

Computer Programming Initial test 11-10-2019
--

1. (2 points) A string can be changed in two ways: either append its last character, or append its length modulo 10, as a digit. Write a function `void printAllStrings(char *s1, unsigned char k);` that takes a positive k and prints all strings that can be obtained from strings denoted by `s1` after making exactly k changes.
2. (2 points) Implement a function `unsigned long getMaxRightShiftRotate(unsigned long x);` which returns the maximum number which can be obtained from x by shifting nibbles of x to the right, with rotation.
3. (3 points) Write a function `double* copyFloats(const char *s, unsigned *n);` that returns an array of doubles, consisting of all the float constants found into string s . Consider floats as being written in any acceptable C format. Set value pointed by n to the number of elements from the returned array. processed
4. (3 points) Write a program that reads a file named on the command line, containing lines of the form `identifier=any-char-string`. Print the lines sorted alphabetically by identifier. Lines have at most 80 chars. Identifiers are like in C. Ignore incorrect lines.

Computer Programming
1st. Attempt; Variant A
29-02-2020

1. (10 points) Implement a function which reads characters from input until EOF and for those lines whose contents consists of exactly three integer numbers, between 1 and 5, separated by at least one white-space character (as given by `isspace()`) prints the numbers, in increasing order, and their average value.
I.e.: a possible such line would be `anna has 2 apples, 4 pears and 3 oranges` which yields `2 3 4 3`
2. (10 points) Implement a function that checks if all nibbles of an `unsigned` (given as parameter) have non-decreasing values, from most to least significant. Use bitwise operators.
I.e.: `f(0x6AD)=1, f(0x243)=0`.
3. (10 points) Implement a function that fills an array of `doubles` (passed as parameter to the function, along with it's capacity) from a string of whitespace-separated `floats` (in any valid C format). Copy as many values as possible, without exceeding the capacity and return the number of copied values.
4. (10 points) Write a function that prints in alphabetical order all headings from an HTML file named on the command line. A heading is enclosed between tags `<hN>` and `</hN>` with N a digit from 1 to 6. Headings are at most 100 chars long, but arbitrarily many.
I.e.: a possible input is `<h2>Upt</h2> <h3>Acsa</h3>` which would print `Acsa Upt`

Notes

- Stick to the problem statement, asking for clarifications when in doubt.
- Problems 1 and 2 are accounted for 1st. part and problems 3 and 4 for the second part.
- Each problem shall be solved in a distinct C file representing a full working program: you will have to implement the `main()` function
- Name your files like: `NameSurname_B1.c` where `Name` and `Surname` are replaced accordingly, and the number is the number of the problem you are solving.

Computer Programming
1st. Attempt; Variant B
29-02-2020

1. (10 points) Implement a function which reads characters from input until EOF and counts how many lines have at least 3 integer numbers, which are in increasing order. Numbers are separated by at least one white-space character (as given by `isspace()`) except at the beginning or end of the line.
I.e.: a possible such line would be `anna has 24 apples, 412 pears and 3624 oranges`
2. (10 points) Implement a function that checks if all nibbles of an `unsigned` (given as parameter) symmetrical to the *middle* of the number are identical. Use bitwise operators.
I.e.: `f(0x6ADDA60)=1, f(0x243342A)=0`.
3. (10 points) Implement a function that takes an array of `floats` with its number of elements, a character buffer-array with its capacity and builds into the buffer-array a string consisting of all the number separated by comma, without exceeding the capacity (copy as much as possible).
4. (10 points) Write a function that prints in increasing order of their URL length all links from an HTML file named on the command line. An HTML link is of the form `Click` the URL portion is the one enclosed in quotes. URLs are at most 100 chars long, but arbitrarily many.
I.e.: a possible input is `Click1 Click2` which would print `Click2 Click1`

Notes

- Stick to the problem statement, asking for clarifications when in doubt.
- Problems 1 and 2 are accounted for 1st. part and problems 3 and 4 for the second part.
- Each problem shall be solved in a distinct C file representing a full working program: you will have to implement the `main()` function
- Name your files like: `NameSurname_B1.c` where `Name` and `Surname` are replaced accordingly, and the number is the number of the problem you are solving.

Computer Programming
1st. Attempt; Variant C
29-02-2020

1. (10 points) Implement a function which reads characters from input until EOF and for those lines whose contents consists of exactly three integer numbers, between 10 and 20, separated by at least one white-space character (as given by `isspace()`) prints the numbers, in increasing order, and their average value.
I.e.: a possible such line would be `anna has 20 apples, 14 pears and 13 oranges` which yields `13 14 20 15.6`
2. (10 points) Implement a function that checks if all nibbles of an `unsigned` (given as parameter) have strictly increasing values, from most to least significant. Use bitwise operators.
I.e.: `f(0x6AD)=1, f(0x243)=0`.
3. (10 points) Implement a function that fills an array of `floats` (passed as parameter to the function, along with it's capacity) from a string of comma-separated `floats` (in any valid C format). Copy as many values as possible, without exceeding the capacity and return the number of copied values.
4. (10 points) Write a function that prints in length increasing order all headings from an HTML file named on the command line. A heading is enclosed between tags `<hN>` and `</hN>` with `N` a digit from 1 to 6. Headings are at most 100 chars long, but arbitrarily many.
I.e.: a possible input is `<h2>Upt</h2> <h3>Acsa</h3>` which would print `Upt Acsa`

Notes

- Stick to the problem statement, asking for clarifications when in doubt.
- Problems 1 and 2 are accounted for 1st. part and problems 3 and 4 for the second part.
- Each problem shall be solved in a distinct C file representing a full working program: you will have to implement the `main()` function
- Name your files like: `NameSurname_C1.c` where `Name` and `Surname` are replaced accordingly, and the number is the number of the problem you are solving.

Computer Programming
1st. Attempt; Variant D
29-02-2020

1. (10 points) Implement a function which reads characters from input until EOF and counts how many lines have at exactly 3 integer numbers, between 1 and 8, which are in increasing order. Numbers are separated by at least one white-space character (as given by `isspace()`) except at the beginning or end of the line.
I.e.: a possible such line would be `anna has 2 apples, 3 pears and 6 oranges`
2. (10 points) Implement a function that checks if all nibbles of an `unsigned` (given as parameter) symmetrical to the *middle* of the number are identical. Use bitwise operators.
I.e.: `f(0x6ADDA60)=1, f(0x243342A)=0`.
3. (10 points) Implement a function that takes an array of `floats` with its number of elements, a character buffer-array with its capacity and builds into the buffer-array a string consisting of all the numbers separated by semicolon, without exceeding the capacity (copy as much as possible).
4. (10 points) Write a function that prints in lexicographical order of the URLs all the links from an HTML file named on the command line. An HTML link is of the form `Click` the URL portion is the one enclosed in quotes. URLs are at most 100 chars long, but arbitrarily many.
I.e.: a possible input is `Click1 Click2` which would print `acsa.upt.ro www.upt.ro`

Notes

- Stick to the problem statement, asking for clarifications when in doubt.
- Problems 1 and 2 are accounted for 1st. part and problems 3 and 4 for the second part.
- Each problem shall be solved in a distinct C file representing a full working program: you will have to implement the `main()` function
- Name your files like: `NameSurname_D1.c` where `Name` and `Surname` are replaced accordingly, and the number is the number of the problem you are solving.

Computer Programming
2nd. Attempt; Variant A
14-02-2020

1. (10 points) Implement a function which reads from standard input data organized as CSV (comma separated values) and for each line prints how many of the fields consists only of alphanumerical characters (as given by `isalpha()`).
2. (10 points) Implement a function which receives an `uint16_t a` number and another `uint64_t b` and checks whether all the bytes of `a` are to be found into `b` (in the same order but not necessarily adjacent)
I.e. `a=0xAB12` and `b=0xAB32DFF1278` is a pair of such numbers;
3. (10 points) Implement a function that receives an array of `uint8_t`, together with it's number of elements, another array of characters together with it's capacity and builds into the second array a "sentence" consisting of the words made of the printable characters (as given by `isprint()`), separated by spaces associated to the non printable chars from the initial array (a sequence of non printable characters is going to be replaced with a single space).
4. (10 points) Write a program that reads from the file given as command line argument a set of floating point numbers (in any C format), one set per line, separated by at least a space (as given by `isspace()`). The program prints the sets, sorted by the number of digits of the decimal part of the numbers and breaks ties by the average value of the data. A data set (line) might have up to 100 numbers, but there might be an arbitrary number of sets (lines)

Notes

- Stick to the problem statement, asking for clarifications when in doubt.
- Problems 1 and 2 are accounted for 1st. part and problems 3 and 4 for the second part.
- Each problem shall be solved in a distinct C file representing a full working program: you will have to implement the `main()` function
- Name your files like: `NameSurname_B1.c` where **Name** and **Surname** are replaced accordingly, and the number is the number of the problem you are solving.

Computer Programming
2nd. Attempt; Variant B
14-02-2020

1. (10 points) Implement a function which reads from standard input data organized as a variant of CSV, where the fields are separated by semicolon (;) and for each line prints the maximum of the negative integers on that line. There might be empty fields, other data than numbers and other kinds of numbers. Those are to be ignored.
2. (10 points) Implement a function which receives an `uint16_t a` number and another `uint64_t b` and checks whether all the nibbles of `a` are to be found into `b` (in the same order but not necessarily adjacent)
I.e. `a=0xAB12` and `b=0xAD2B32DFF1C278` is a pair of such numbers;
3. (10 points) Implement a function that receives a string representing an arithmetic expression consisting only of additions and subtractions of floating point numbers (in any C format) and returns the values of the expression. Deviations from the format shall be handled as errors.
4. (10 points) Write a program that reads from the file given as command line argument pairs of values of the format `key=value`, where `key` and `value` are strings, and for those lines having the `key` equal to `upt` sorts the pairs in ascending order of their values (if they are numerical) or by their length (if they are not numbers). There might be an arbitrary number of sets (lines), while `key` and `value` are smaller than 100 characters .

Notes

- Stick to the problem statement, asking for clarifications when in doubt.
- Problems 1 and 2 are accounted for 1st. part and problems 3 and 4 for the second part.
- Each problem shall be solved in a distinct C file representing a full working program: you will have to implement the `main()` function
- Name your files like: `NameSurname_B1.c` where **Name** and **Surname** are replaced accordingly, and the number is the number of the problem you are solving.

Computer Programming
2nd. Attempt; Variant C
14-02-2020

1. (10 points) Implement a function which reads characters from input until EOF and for those lines whose contents consists of exactly three integer numbers, between 10 and 20, separated by at least one white-space character (as given by `isspace()`) prints the numbers, in increasing order, and their average value.
I.e.: a possible such line would be `anna has 20 apples, 14 pears and 13 oranges` which yields `13 14 20 15.6`
2. (10 points) Implement a function that checks if all nibbles of an `unsigned` (given as parameter) have strictly increasing values, from most to least significant. Use bitwise operators.
I.e.: `f(0x6AD)=1, f(0x243)=0`.
3. (10 points) Implement a function that fills an array of `floats` (passed as parameter to the function, along with it's capacity) from a string of comma-separated `floats` (in any valid C format). Copy as many values as possible, without exceeding the capacity and return the number of copied values.
4. (10 points) Write a function that prints in length increasing order all headings from an HTML file named on the command line. A heading is enclosed between tags `<hN>` and `</hN>` with `N` a digit from 1 to 6. Headings are at most 100 chars long, but arbitrarily many.
I.e.: a possible input is `<h2>Upt</h2> <h3>Acsa</h3>` which would print `Upt Acsa`

Notes

- Stick to the problem statement, asking for clarifications when in doubt.
- Problems 1 and 2 are accounted for 1st. part and problems 3 and 4 for the second part.
- Each problem shall be solved in a distinct C file representing a full working program: you will have to implement the `main()` function
- Name your files like: `NameSurname_C1.c` where `Name` and `Surname` are replaced accordingly, and the number is the number of the problem you are solving.

Computer Programming
2nd. Attempt; Variant D
14-02-2020

1. (10 points) Implement a function which reads characters from input until EOF and counts how many lines have at exactly 3 integer numbers, between 1 and 8, which are in increasing order. Numbers are separated by at least one white-space character (as given by `isspace()`) except at the beginning or end of the line.
I.e.: a possible such line would be `anna has 2 apples, 3 pears and 6 oranges`
2. (10 points) Implement a function that checks if all nibbles of an `unsigned` (given as parameter) symmetrical to the *middle* of the number are identical. Use bitwise operators.
I.e.: `f(0x6ADDA60)=1, f(0x243342A)=0`.
3. (10 points) Implement a function that takes an array of `floats` with its number of elements, a character buffer-array with its capacity and builds into the buffer-array a string consisting of all the numbers separated by semicolon, without exceeding the capacity (copy as much as possible).
4. (10 points) Write a function that prints in lexicographical order of the URLs all the links from an HTML file named on the command line. An HTML link is of the form `Click` the URL portion is the one enclosed in quotes. URLs are at most 100 chars long, but arbitrarily many.
I.e.: a possible input is `Click1 Click2` which would print `acsa.upt.ro www.upt.ro`

Notes

- Stick to the problem statement, asking for clarifications when in doubt.
- Problems 1 and 2 are accounted for 1st. part and problems 3 and 4 for the second part.
- Each problem shall be solved in a distinct C file representing a full working program: you will have to implement the `main()` function
- Name your files like: `NameSurname_D1.c` where `Name` and `Surname` are replaced accordingly, and the number is the number of the problem you are solving.