

Exercises and assignments, C-programming

Work package nr 2 / General C-programming

(Max 12 p)

Exerc_2_1 : (Filename exerc_2_1.c)

(1p)

Write a program that reads in a string with a maximum of 20 characters from the keyboard and stores the string in a local string variable.

Copy the string to another string by using:

a) The library function `strcpy(..)`

b) Your own function ***void copyString(...)*** not using any library function.

Main program ends by printing out the copied string in the console.

The program should be able to both read in from keyboard or from a text file 'myfile.txt' containing one string of characters. You create this file with notepad or any other text editor. The reading from the text file should be done by redirect the readings from command line when program execution starts as follows: *Exerc_2_1 < myfile.txt*

Where Exerc_2_1 is the filename of the compiled program. You **shall not** use standard file managing by opening the file and the read from it.

Exerc_2_2: (Filename exerc_2_2.c)

(1p)

Write a program that creates an array of integers, `array[MAX]`, and then fill it with MAX no of random integers from 1 to 99. Let then the program prints out the following:

The value of the label *array* (address) is: xxxxxxxxxx

First integer in the array is (`array[0]`) : xxxxxxxxxx

The size of an integer (number of bytes) is : xxxxxxxxxx

The size of the whole array is : xxxxxxxxxx

The program shall then, by use of a pointer, print out each integer value and it's double value.

Exerc_2_3 : (Filename exerc_2_3.c)

(1p)

Create a program that reads in a number of strings **from the command line** when it starts and then checks if it is two strings and if exactly two check if they are identical or not. Do this with and without use of library function `strcmp(..)`. Let the program print out the result in some way.

Start from command line: *exerc_2_3 string1 string2*

Exerc_2_4 : (Filename exerc_2_4.c)

(1p)

Create a program that reads in a string and determines if the string is a palindrome. A word is a palindrome if it is the same word reading from left to right or right to left. We can assume that it is a simple strings without any space character in it.

Example of palindrome : level, rotor and racecar.

Exerc_2_5: Pointer exercise

(2p)

Exercises and assignments, C-programming

You should develop a program that calculates some statistical values for an array of integers. Among other things, the program will plot a histogram for the frequency of different numbers in the array. To test it you need to create an array of integers (*table* [MAX]) with MAX number of random numbers between 0 and MAXNUMBER. Then you should write a function that for each possible number between 0 – MAXNUMBER calculates how many times the number exists in the array. The result is then stored in a new array (*frequency* []).

Finally, you write a function that given the array *frequency* [] draw a histogram as below figure:

You should use the function declaration as below.

Given an array *table*[]={ 1,2,12, 5,1,0,0,5,9,12, 0,2,3,0} the program will printout:

```
0  xxxx
1  xx
2  xx
3  x
5  xx
9  x
12 xx
```

Note: Numbers with frequency 0 in the array *frequency*[] is not printed out.

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

```
#include <stdlib.h>
```

```
#include <time.h>
```

```
#define MAX 100
```

```
#define MAXNUMBER 20
```

```
// ----- Function declaration -----
```

```
void create_random( int *tab );           // Use pointer to fill the table
```

```
void count_frequency(int *tab, int *freq ); // Use pointer
```

```
void draw_histogram(int *freq );         // Use pointer
```

```
int main ( void){
```

```
    int table[MAX], n ;
```

```
    int frequency[MAXNUMBER];
```

```
    ....
```

```
}
```

Exercises and assignments, C-programming

We want to use an array for a queue of positive integers. The numbers must be entered in to the queue which is done by putting them into the first vacant location in the field. (In the figure below from left) . An integer is taken out from the queue by taking the first number in the array. When an integer is taken out the other integers should be shifted one step up to left in the queue.

A vacancy in the queue is represented by the integer -1. In a full queue there is no integer -1.

When taking out an integer from a full queue , the integer -1 must be written in the last position.

Examples below show a queue with five positions.

```
int que [ MAX ] ; // MAX equal to 5 in this example
```

Queue from the start:	-1 -1 -1 -1 -1
Queue after you first put in the numbers 3 and then 4 and 5.	3 4 5 -1 -1
Queue after additional put in of 8 :	3 4 5 8 -1
And after taken out an integer	4 5 8 -1 -1

a) Write a function void ***initQue (int list [], int max)*** that initializes a list to initially containing max nr of vacant positions (-1).

b) Write a function ***int input (int list [], int number, int max)*** that adds a number in the queue according to the rules , and return 1 if the number could be entered and 0 if the queue is full.

c) Write a function ***int output(int list[], int max)*** that returns the value if any to fetch or 0 if que is empty.

d) Finally write a main-program that sets up an empty queue with 5 positions and then call all functions for test of its function.

Exerc_2_7

(2p)

Create a program that checks an entered Swedish person number. The number is entered in the form of : 7107254786 (yymmddxxxc). The number should be read in as a string and converted to integers for year, month , day and number.

The last digit (here 6) is a control digit and is calculated from the other digits by an algorithm that you can find on the internet.

The user inputs the number, the program first checks that the number of month and day is in the right range and after that calculate and checks the control digit. The program then prints out the result and asks for a new person number. This is repeated until the user inputs a 'q'.

The program should at least consist of the functions:

main() , readPersnr(char *person), int controlDigit(const char * persnr) .

Exercises and assignments, C-programming

Exerc_2_8

(2p)

a) The game Nim works as follows: On the table is a stack of 13 coins. Two players take in turn between one and three coins from the stack. Whoever are forced to take the last coin have lost.

Your task is to develop the game Nim by using the code skeleton which is available on the course webpage in Documents / Work package nr 2 .

The program is also available in the form of an executable demo program at the same place.

- Download the code skeleton. Read the code and the comments, and try to understand how the program is structured.
- Compile and run it.
- In the function declarations you can read a description of what the function should do. All function definitions are empty (called stubs). Read through the comments and develop the functions so that the program works. NOT all at once, do ONE by ONE. Try to test each function in any way. For example use the debugger function or print out a value.

TIPS: Create a test area at the beginning of main. Test there to call and print the result of a function at a time. You can write `return (0)` immediately after printing, so you do not run the entire program. Remove the test parts when the test is complete.

b) When the game works well, you can only play one round. Change the program so you can play several rounds. After each round the program asks if you want to play again. Use function `play again()`
