Technical University of Denmark

Technical University of Denmark

Written examination, December 9, 2020

Course title:	Mathematical Software Programming
Course number:	02635
Aids allowed:	All aids allowed
Exam duration:	4 hours
Weighting:	80/100

Final exam
Mathematical Software Programming

This exam contains a total of 18 questions: 14 multiple choice questions (questions 1–14) and 4 programming questions (questions 15–18). The ZIP file attached to the exam contains a text document for your multiple choice answers and templates for the programming questions. You must use these files for your answers.

---

1. (4 points) Properties of floating-point numbers.

   (a) Can the number 1/3 be represented as a finite precision floating-point number in base 10?

      A. Yes

      B. No

   (b) Can the number 1/3 be represented as a finite precision floating-point number in base 2?

      A. Yes

      B. No

2. (2 points) Consider the following code snippet:

```c
double x = 33.0/395.0, y = 1.0/12.0;
double z = x-y;
```

   How many significant binary digits (i.e., bits) are lost in the subtraction $z = x - y$?

   A. At least 6 bits and at most 7 bits.

   B. At least 7 bits and at most 8 bits.

   C. At least 8 bits and at most 9 bits.

   D. At least 9 bits and at most 10 bits.

3. (2 points) Recall that binary64 is a binary floating-point format with precision 53. How many distinct binary64 floating-point numbers are there in the interval $[1, 2)$?

   A. 2^{51}

   B. 2^{52}

   C. 2^{53}

   D. 2^{54}

4. (2 points) Consider the following representation of a floating-point number in base b

$$(d_0.d_1d_2\dots d_{p-1})_b \cdot b^E.$$

When is this representation called *normal*?

   A. When the precision p is an even number.

   B. When the exponent E is positive.

   C. When the exponent E is zero.

   D. When the leading digit, d_0, is nonzero.

   E. When the base b is equal to 2.

5. (4 points) The theoretical improvement in speed of execution of a task executed on p processors can be expressed as

$$S(p) = \frac{T(1)}{T(p)} = \frac{fT(1) + (1-f)T(1)}{(f/p)T(1) + (1-f)T(1)}$$

where $T(p)$ is the execution time on p processors (real time) and f is the so-called parallel fraction of the task. For example, if 60% of a task can be parallelized, then $f = 0.6$.

   (a) Now suppose you have a computer with $p = 8$ processors and a task whose parallel fraction is $f = 0.95$. Is it possible to achieve a speed-up of 6 by exploiting all 8 processors?

      A. Yes

      B. No

   (b) At least how many of the 8 processors are needed to achieve a speed-up of 4?

      A. 4

      B. 5

      C. 6

      D. 7

      E. 8

6. (2 points) What is the *stride* of an array?

   A. The number of elements in the array.

   B. The size of the array in bytes.

   C. The distance in memory between consecutive elements of the array.

   D. The data type of the elements of the array.

7. (4 points) Suppose the matrix

$$C = \begin{bmatrix} C_{1,1} & C_{1,2} & C_{1,3} & C_{1,4} & C_{1,5} \\ C_{2,1} & C_{2,2} & C_{2,3} & C_{2,4} & C_{2,5} \\ C_{3,1} & C_{3,2} & C_{3,3} & C_{3,4} & C_{3,5} \end{bmatrix}$$

is stored as a one-dimensional, row-major `double` array of length 15, and let the variable `pC` be a pointer to the first element of C (i.e., the variable `pC` stores the address of $C_{1,1}$).

(a) Which of the following expressions is a pointer to the element $C_{2,2}$?

A. `pC+4`
B. `pC+5`
C. `pC+6`
D. `pC+7`

(b) Now suppose that we wish to carry out the operation

$$\begin{bmatrix} C_{2,1} \\ C_{3,1} \end{bmatrix} \leftarrow \begin{bmatrix} C_{2,2} & C_{2,3} & C_{2,4} \\ C_{3,2} & C_{3,3} & C_{3,4} \end{bmatrix} \begin{bmatrix} C_{1,5} \\ C_{2,5} \\ C_{3,5} \end{bmatrix}$$

using the BLAS function DGEMV which performs matrix-vector operations of the form $y \leftarrow \alpha Ax + \beta y$ and $y \leftarrow \alpha A^T x + \beta y$. Which of the following expressions should be passed to DGEMV as a pointer to the first element of the vector y?

A. `pC+4`
B. `pC+5`
C. `pC+6`
D. `pC+7`

8. (4 points) Consider the function $f(x) = \int_0^1 e^{tx}\,dt$ which can also be expressed as

$$f(x) = \begin{cases} 1, & x = 0, \\ \frac{e^x-1}{x}, & x \neq 0. \end{cases}$$

(a) The problem of evaluating f when x is near 0 is

A. well-conditioned.

B. ill-conditioned.

C. neither well-conditioned nor ill-conditioned; it depends on how f is evaluated and/or the numerical precision.

(b) Consider the following implementation of a function to evaluate f numerically:

```
double feval(double x) {
    if (x == 0)
        return 1.0;
    else
        return (exp(x)-1.0)/x;
}
```

Is this a numerically stable algorithm for evaluating f?

A. Yes

B. No

9. (4 points) Consider the following implementation of sequential summation:

```
double sequential_sum(double * arr, int n) {
    double sum = 0.0;
    for (int i=0;i<n;i++)
        sum += arr[i];
    return sum;
}
```

(a) References to the elements of the array `arr` are

A. spatially local.

B. temporally local.

(b) References to the variable `sum` are

A. spatially local.

B. temporally local.

10. (2 points) A reference in C++ is the same as a pointer in C.

    A. True

    B. False

11. (2 points) A class in C++ is

    A. a variable.

    B. an abstract data type.

    C. a mathematical operator.

    D. an address in memory.

12. (4 points) Consider the following recursive definition of a class of polynomials:

$$
\begin{aligned}
H_0(x) &= 1 \\
H_1(x) &= 2x \\
H_n(x) &= 2xH_{n-1}(x) - 2nH_{n-2}(x), \qquad n = 2, 3, \ldots
\end{aligned}
$$

(a) What kind of recursion is this?

   A. Single recursion

   B. Multiple recursion

(b) What is the complexity associated with the following algorithm for evaluating $H_n(x)$?

```
double H(double x, unsigned int n) {
   if (n==0) return 1.0;
   if (n==1) return 2*x;
   return 2*x*H(x,n-1)-2*n*H(x,n-2);
}
```

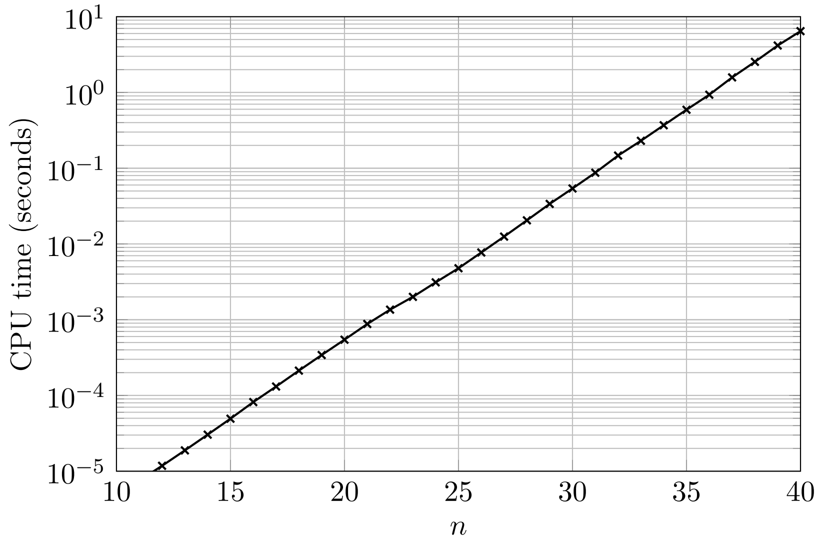
   A. $O(n)$ space and $O(n)$ time

   B. $O(2^n)$ space and $O(n)$ time

   C. $O(n)$ space and $O(2^n)$ time

   D. $O(2^n)$ space and $O(2^n)$ time

13. (4 points) The following plot shows the CPU time required by some algorithm to solve a certain problem as a function of its dimension n.



What is the time complexity associated with the algorithm?

A. $O(\log n)$

B. $O(n)$

C. $O(n^\alpha), \alpha > 1$

D. $O(2^n)$

14. (4 points) Memory management.

(a) A cache miss is the result of a memory leak.

A. True

B. False

(b) Consider the following expression:

```
int myArray[64];
```

Which kind of memory allocation will be used to allocate the array `myArray`?

A. Automatic allocation

B. Dynamic allocation

15. (8 points) Implement a function in C that evaluates the function

$$f(x) = \frac{1 - \cos(x)}{\sin(x)}.$$

Requirements

- Your function must have the following prototype:

```
double feval(double x);
```

- The function must not write anything to `stdout`.
- Use the template `exam_e20_question15.c` for your implementation. The template is included in the ZIP file attached to the exam.

16. (8 points) Implement a function in C that evaluates

$$g(x) = \sqrt[3]{x^2 + 2} - \sqrt[3]{x^2}, \quad x \in \mathbb{R}.$$

Hint: Use the identity

$$(a - b)(a^2 + ab + b^2) = a^3 - b^3$$

to rewrite $g(x)$ in order to avoid loss of significance for certain values of x.

Requirements

- Your function must have the following prototype:

```
double geval(double x);
```

- The function must not write anything to `stdout`.
- Use the template `exam_e20_question16.c` for your implementation. The template is included in the ZIP file attached to the exam.

17. (8 points) Let $x = (x_1, \ldots, x_n)$ and $w = (w_1, \ldots, w_n)$ be two vectors of length n with positive elements. The function

$$H(x; w) = \frac{\sum_{i=1}^{n} w_i}{\sum_{i=1}^{n} \frac{w_i}{x_i}}$$

is a weighted harmonic mean of $x_1, \ldots, x_n$ with weights $w_1, \ldots, w_n$..

Implement a function in C that evaluates $H(x; w)$.

Requirements

- Your function must have the following prototype:

```c
double weighted_harmonic_mean(int n, double * x, double * w);
```

  The input `n` is the length of the vectors x and w, `x` is a pointer to the first element of the vector x, and `w` is a pointer to the first element of w.
- The function should return $H(x; w)$ if the input is valid, and otherwise it should return `NAN`.
- The function must not write anything to `stdout`.
- Use the template `exam_e20_question17.c` for your implementation. The template is included in the ZIP file attached to the exam.

18. (12 points) Let L be a square matrix of order n and of the form

$$L = \begin{bmatrix} a_1 & & & & \\ & a_2 & & & \\ & & \ddots & & \\ & & & a_{n-1} & \\ b_1 & b_2 & \cdots & b_{n-1} & a_n \end{bmatrix},$$

i.e., L is a lower-triangular matrix where only elements on the diagonal and in the last row can be nonzero.

(a) Implement a function in C that performs the operation $x \leftarrow Lx$, i.e., the function should overwrite the input array x by Lx.

Requirements

- Your function must have the following prototype:

```c
int darmv(int n, double * a, double * b, double * x);
```

  The input `n` is the order of the matrix L. The inputs `a`, `b`, and `x` are pointers to the first elements of the arrays that represent the vectors $a = (a_1, \dots, a_n)$, $b = (b_1, \dots, b_{n-1})$, and $x = (x_1, \dots, x_n)$, respectively.
- The function should return the value -1 in case of an invalid input, and otherwise it should return the value 0.
- The function must not write anything to `stdout`.
- Use the template `exam_e20_question18a.c` for your implementation. The template is included in the ZIP file attached to the exam.

(b) Implement a function in C that performs the operation $x \leftarrow L^{-1}x$, i.e., the function should overwrite the input array x by $L^{-1}x$.

Requirements

- Your function must have the following prototype:

```c
int darsv(int n, double * a, double * b, double * x);
```

  The input `n` is the order of the matrix L. The inputs `a`, `b`, and `x` are pointers to the first elements of the arrays that represent the vectors $a = (a_1, \dots, a_n)$, $b = (b_1, \dots, b_{n-1})$, and $x = (x_1, \dots, x_n)$, respectively.
- The function should return the value -1 in case of an invalid input or if L is singular, and otherwise it should return the value 0.
- The function must not write anything to `stdout`.
- Use the template `exam_e20_question18b.c` for your implementation. The template is included in the ZIP file attached to the exam.