Introduction to Languages for Scientific Computing, winter semester 14/15: Final Exam

Lecture given by: Paolo Bientinesi

First exam, 10.02.2015

The following document is a transcript from memory created by some students. There is no guarantee for correctness or completeness. Especially concrete values may differ. However, the conceptual design should be correct. Please notice that this document is not published by any member of professor Bientinesi's group.

1) Floating point arithmetic

1.1) [6 points]

Consider the following floating point arithmetic with normalization:

$$\beta = 2, \quad t = 4, \quad e_{min} = -5, \quad e_{max} = 5.$$

1. What is $a = fl(12.34)$?
2. What is the absolute and relative representation error of a?
3. What is $b = fl(0.321)$?
4. What is the absolute and relative representation error of b?
5. What is $c = fl(a + b)$?
6. What is the absolute and relative representation error of c?

(One point each).

1.2) [2 points]

Consider single and double precision arithmetic with normalization:

$$\beta = 2, \quad t = 24, \quad e_{min} = -125, \quad e_{max} = 128,$$
$$\beta = 2, \quad t = 53, \quad e_{min} = -1021, \quad e_{max} = 1024.$$

How many double precision numbers are between two consecutive single precision numbers (i.e. how many d_js in $]s_i, s_{i+1}[$)?

2) Matlab

2.1) [2 points]

What is the output of the following program?

```
A=[2 0;0 2];
B=[1 2;-3 4];
M = A'*B
v([4 3 1 2]) = [M(1,:),M(2:)]
```

2.2) [3 points]

What is the output of the following program?

1. `sin(pi)`
2. `2^54 + 2 == 2^54`
3. `r = rand; s = single(r); t = double(s); s == t`

2.3) [4 points]

Write a function that takes a vector of size $3n$ and returns

$$\begin{bmatrix} I_n & I_n & 0 \\ -I_n & 0 & I_n \\ I_n & -I_n & 0 \end{bmatrix} \cdot v$$

without constructing the matrix, or I_n.

2.4) [3 points]

Write a recursive program that constructs H_n:

$$H_n = \begin{bmatrix} n & n & n & n & n & n \\ n & n-1 & n-1 & \dots & n-1 & n-1 \\ n & n-1 & n-2 & \dots & n-2 & n-2 \\ n & n-1 & n-2 & \dots & \vdots & \vdots \\ n & n-1 & n-2 & \dots & 1 & 1 \\ n & n-1 & n-2 & \dots & 1 & 0 \end{bmatrix}$$

3) Mathematica

The constructs `For, While, Do, Table` and `If` are forbidden.

3.1) [2 points]

Explain the following expression

```
f[x__,n_Integer];/Length[{x}]>n -> g[x]
```

3.2) [3 points]

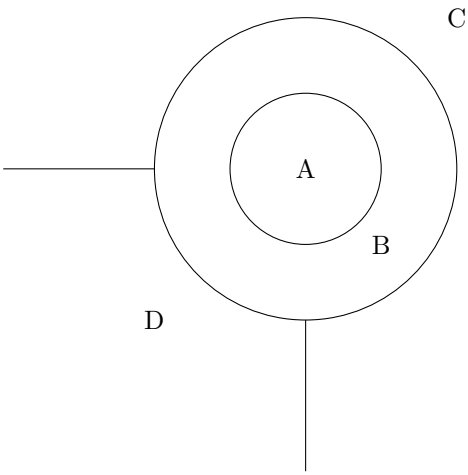
What is the output of the following program?

```
Map[ {#[[1]][#[[2]]], #[[1]][2]#[[2]], #[[2]]#[[1]][#[[1]][1]]},
     {{#^2-1&, 2},{(#-1)(#+1)&, -2}}
  ]
```

3.3) [4 points]

Write a function `H[i,j,n]` that returns the (i, j)-entries of H_n as in 2.4 with bound checks. If out-of-bounds or no integer: return "NaN".

3.4) [2 points]



Function ABC as follows:

- in A (circle size 1 around (0,0), incl. borders): $x^2 + y^2$
- in B (circle size 2 around that, incl. outer borders):cos $\sqrt{x^2 + y^2}$
- in C (outside circle): 0
- in D (outside circle, lower left quarter): π

Define function ABC.

3.5) [5 points]

Captian Kirk wants to find the roots of $\left|\sin\left(x^2\right) - .2\right| - .3$ in $[0, \pi]$. FindRoots takes an equation and a tuple of the form {`variable, initial_guess`} and returns one solution in the form {`variable->value`}. Help him follow these steps:

1. Define f.
2. Apply FindRoots for initial guesses given by `Table[i,{i,-.2,4,.2}]`.
3. Filter those results outside the range.
4. Merge results which are "almost equal".
5. Plot the function as shown in figure ... (Function + Dots at every root).

4) C

4.1) [5 points]

Leia successfully compiles the following program:

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

void function(int* arr, int n){
  int idx, current,i,j,tmp;

  for (i = 0; i < n; i++) {
    current = -1;
    idx = -1;

    for(j=i; j < n; j++) {
      if (arr[j] > current) {
        current = arr[j];
        idx = j;
      }
    }

    printf("iter:%d\tidx:%d\n", i, idx);

    tmp = arr[idx];
    arr[idx] = arr[i];
    arr[i] = tmp;
  }
}
```

```
int main (int argc, char* argv[]) {
  int i, n = 6;

  int* arr = (int*) malloc(n*sizeof(int));

  for (i=0; i < n; i++) {
    arr[i] = (7*i+2) % 5;
  }

  function(arr, n);

  printf("[");

  for (i = 0; i < n; i++) {
    printf("%d␣", arr[i]);
  }

  printf("]");
  return 0;
}
```

1. What is the output of the program?
2. What is generally computed by function?
3. If `for (i = 0; i < n; i++)` is replaced with `for (i = 0; i < n-1; i++)`, would the output differ? Why?
4. If `for (j = i; j < n; j++)` is replaced with `for (j = i+1; j < n; j++)`, would the output differ? Why?

4.2) [3 points]

The Princess wants to make calculations with the class of matrices P_n. Construct lower-triangular matrix $P(i,j) = \frac{i+j}{ij+1}$ for $i < j$, 0 else. Ensure that it is stored by columns.

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

int main(int argc, char* argv[]) {
  int n;
  if(argc >1) n=atoi(argv[1]);

  //your code
}
```