Eidgenössische Technische Hochschule Zürich
Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo
Swiss Federal Institute of Technology Zurich

Institut für Theoretische Informatik
Dr. B. Gärtner, Prof. Dr. J. Hromkovič

8. November 2011

Informatik für Mathematiker und Physiker Serie 7 HS 11

URL: http://www.ti.inf.ethz.ch/ew/courses/Info1_11/

Skript-Aufgabe 78 (4 Punkte)

Assume that in some program, a is an array of underlying type int and length n.

a) Given a variable i of type int with value $0 \leq i \leq n$, how can you obtain a pointer p to the element of index i in a? (Note: if $i = n$, this is asking for a past-the-end pointer.)

b) Given a pointer p to some element in a, how can you obtain the index i of this element? (Note: if p is a past-the-end pointer, the index is defined as n.)

Write code fragments that compute p from i in a) and i from p in b).

Skript-Aufgabe 79 (4 Punkte)

Find and fix at least 5 problems in the following program. The fixed program should indeed correctly do what it claims to do. Is the fixed program const-correct? If not, make it const-correct! (This is a theory exercise, but you may of course use the computer to help you.)

```
#include<iostream>

int main()
{
  int a[7] = {0, 6, 5, 3, 2, 4, 1}; // static array
  int* const b = new int[7];

  // copy a into b using pointers
  for (int* p = a; p <= a+7; ++p)
    *b++ = *p;

  // cross-check with random access
  for (int i = 0; i <= 7; ++i)
    if (a[i] != b[i])
      std::cout << "Oops, copy error...\n";

  delete b;

  return 0;
}
```

Skript-Aufgabe 80 (4 Punkte)

Let us call a natural number k-*composite* if and only if it is divisible by exactly k different prime numbers. For example, prime powers are 1-composite, and $6 = 2 \cdot 3$ as well as $20 = 2 \cdot 2 \cdot 5$ are 2-composite. Write a program k_composite.cpp that reads numbers $n \geq 0$ and $k \geq 0$ from the input and then outputs all k-composite numbers in $\{2, \ldots, n-1\}$. How many 7-composite numbers are there for $n = 1,000,000$?

Skript-Aufgabe 83 (4 Punkte)

Enhance the program `read_array.cpp` from Exercise 82 so that the resulting program `sort_array.cpp` sorts the array elements into ascending order before outputting them. Your sorting algorithm does not have to be particularly efficient, the main thing here is that it works correctly. Test your program on some larger inputs (preferably read from a file, after redirecting standard input). The input is formed as follows. The first integer specifies the number of integers to be sorted. This is followed by that many integer numbers, which are separated by whitespaces. For example, on input `5 4 3 6 1 2` the program should output `1 2 3 4 6`.

Programm: read_array.cpp

```
// Program: read_array.cpp
// read a sequence of n numbers into an array
#include <iostream>

int main()
{
  // input of n
  unsigned int n;
  std::cin >> n;

  // dynamically allocate array
  int* const a = new int[n];

  // read into the array
  for (int i=0; i<n; ++i) std::cin >> a[i];

  // output what we have
  for (int i=0; i<n; ++i) std::cout << a[i] << " ";
  std::cout << "\n";

  // delete array
  delete[] a;

  return 0;

}
```

Die **Aufgaben 74** und **88** aus den Vorlesungsunterlagen sind die **Challenge Aufgaben** und geben jeweils 8 Punkte.

Abgabe: Bis 15. November 2011, 15.15 Uhr.