Fondamenti di Informatica 1 - Compito A

Prof. Marco Gavanelli

16 June 2003

Exercise 1 (4 pts)

A computer represents integer numbers in two's complement representation with 4 bits. It represents real numbers in floating point with 5 bits for the normalized mantissa and 4 bits for the exponent.

Show how such a computer performs the following calculation:

$$1,34 + (14 - 1)$$

Show the internal representation of the result. Highlight the errors and show the result both in decimal and in binary.

Exercise 2 (13 pts)

A binary file PILOTS.DAT contains information about the arrival order of pilots in Formula 1 races. For each pilot are recorded the following data: surname (a string of 20 chars) and an array of 20 integers. The element of index *i* represents the arrival position of the pilot in the *i*-th race of the championship.

Write a program in C that reads the file PILOTS.DAT and creates an array containing, for each pilot, the surname and the total score at the end of the championship. In order to compute the total score, the program **must contain** a function that takes as input the array with the arrival position in the races and returns the total score. A function `race_score` is provided in a library `score.h`. The interface is the following:

```
int race_score(int);
```

This function takes as input the arrival position of a driver (in a race) and returns the score of the pilot (for that race).

Finally, the program should print the array containing the names of the drivers and their total score.

Exercise 3 (9 pts)

Write a recursive C function with the following interface:

```
int d(int n, int k)
```

that computes the following formula

$$d(n, k) = \prod_{i=-n}^{n} (k + n)$$

Then, show how the following program (invoking the previously defined function) works by means of the activation records:

```
int h(int x, int *y, int z[])
{  int a;
   int c[] = {1,2,3};
   for (a=x; a< (*y); a++)
       z[a]=c[2-a];
   x++; *y= *y +1;
   return d(x,*y);
}

main()
{  int a=1, b=3, c=7, x[3]={0,0,0};
   c=h(a,&b,x);
}
```

Solution 1

Conversion in base 2:

$$1,34_{DEC} = 1,01010\ldots_2 = 0,10101 \cdot 2^{0000}$$

Truncation Error. The value is internally represented as

$$\text{Mantissa} \quad \boxed{0}\boxed{0101} \quad \text{Exponent} \quad \boxed{0000}$$

$$14_{DEC} = 1110_2$$

$$1_{DEC} = 0001_2$$

1's Complement: 1110
2's Complement: 1111

$$\begin{array}{rrc} (14_{DEC}) & 1110 & + \\ (-1_{DEC}) & 1111 & = \\ \hline (13_{DEC}) & (1)1101 & \end{array}$$

Conversion in floating point

$$1101 \cdot 2^0 = 0,11010 \cdot 2^{0100}$$

$$\begin{array}{rlll} (13) & 0,11010 & \cdot 2^{0100} & + \\ (0,85) & 0,00010 & \cdot 2^{0100} & = \quad \textit{Columning error} \\ \hline & 0,11100 & \cdot 2^{0100} & \end{array}$$

The result is in normal form. The internal representation is:

$$\text{Mantissa} \quad \boxed{0}\boxed{1100} \quad \text{Exponent} \quad \boxed{0100}$$

This value is 14_{DEC}. The correct result is 14.34, so the total error is 0.34.

Solution 2

```
#include "score.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define NUMPILOTI 50

int calc_punt(int A[])
{  int i,tot;
```

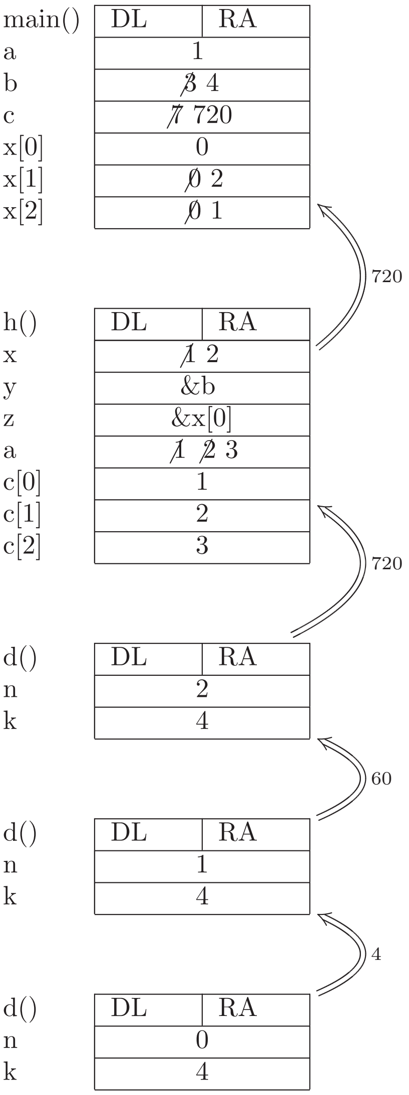
```
   for (i=0; i<20; i++)
       tot += race_score(A[i]);
   return tot;
}

main()
{    int letto,i;

   struct gare_piloti
   { char cognome[20];
     int gara[20]; } p;

   struct punti_piloti
   {char cognome[20];
   int punti;} piloti[NUMPILOTI];

   FILE * fp = fopen("PILOTI.DAT","rb");
   if (fp == NULL)
       exit(1);

   i=0;
   do
   {   letto = fread(&p,1,sizeof(struct gare_piloti),fp);
       if (letto)
       {   strcpy(piloti[i].cognome,p.cognome);
           piloti[i].punti = calc_punt(p.gara);
           i++;
       }
   }
   while (letto);

   fclose(fp);
}
```

Solution 3

Code of the function:

```
int d(int n, int k)
{ if (n==0)
    return k;
    else return (n+k)*(-n+k)*d(n-1,k);
}
```

Activation records:



main()	DL	RA
a	1	
b	~~3~~ 4	
c	~~7~~ 720	
x[0]	0	
x[1]	~~0~~ 2	
x[2]	~~0~~ 1	

h()	DL	RA
x	~~1~~ 2	
y	&b	
z	&x[0]	
a	~~1~~ ~~2~~ 3	
c[0]	1	
c[1]	2	
c[2]	3	

d()	DL	RA
n	2	
k	4	

d()	DL	RA
n	1	
k	4	

d()	DL	RA
n	0	
k	4	