Relazione ESERCITAZIONE SUGLI ERRORI

I componenti del gruppo in ordine alfabetico e i rispettivi numeri di matricola:

Francesco Curcio S4863740

Giacomo Pedemonte S4861715

ESERCIZIO 1:

1. Si consideri il numero di matricola del primo componente, in ordine alfabetico, del gruppo; si indichi con d_0 e d_1, rispettivamente, l'ultima e la penultima cifra di tale numero di matricola.
   Posto $a = (d_0 + 1) \cdot 10^i$, con $i = 0, 1, ..., 6$, $b = (d_1 + 1) \cdot 10^{20}$, $c = -b$, eseguire i seguenti calcoli in aritmetica di macchina a doppia precisione, cioè utilizzando variabili di tipo double:
   - $(a + b) + c$
   - $a + (b + c)$

Svolgimento:

inizializziamo d0 = 4 e d1 = 0 e prendiamo a, b e c come variabili di tipo double (a doppia precisione):

```
int d0 = 0;
int d1 = 4;
double a, b, c;
```

Ora andiamo ad esaminare i risultati ottenuti (sulla destra) dal codice(sulla sinistra):

```
for (int i = 0; i <= 6; i++)
{
    a = (d0 + 1) * pow(10, i);
    b = (d1 + 1) * pow(10, 20);
    c = -b;

    //(a+b)+c
    double res = (a + b) + c;
    cout << "(a+b)+c = " << res << endl;

    //a+(b+c)
    double res2 = a + (b + c);
    cout << "a+(b+c) = " << res2 << endl;

    cout << endl;
}
```

```
Esercizio 1:
(a+b)+c = 0
a+(b+c) = 1

(a+b)+c = 0
a+(b+c) = 10

(a+b)+c = 96
a+(b+c) = 100

(a+b)+c = 992
a+(b+c) = 1000

(a+b)+c = 9984
a+(b+c) = 10000

(a+b)+c = 100000
a+(b+c) = 100000

(a+b)+c = 1e+006
a+(b+c) = 1e+006
```

Nell'output dell'esercizio 1 notiamo che per valori di a piccoli, ovvero nei primi cicli, il risultato di viene approssimato, quindi risulta che: a + b = b .
Questo perché a e b nei primi 5 cicli differiscono di un ordine di grandezza maggiore della precisione di macchina; per questo risulta che: (a+b)+c != a+(b+c)

Notiamo inoltre che in questo esercizio stiamo usando double, quindi doppia precisione, quindi un ordine di grandezza di circa 10^{-16}.

L'algoritmo dell'esercizio 1 funziona meglio quando passiamo alla quinta iterazione, dove a differisce rispetto a b e c di un ordine di grandezza inferiore rispetto a 10^{-16} .

Infatti: Alla quinta iterazione, quando ho i = 4
a = 50000 = $5 \cdot 10^4$
b = $1 * 10^{20}$

Ottenendo la seguente esecuzione:

```
(a+b)+c = 9984
a+(b+c) = 10000
```

I due dati differiscono per quanto detto sopra, quindi da qui in poi non notiamo più approssimazioni così drastiche e l'algoritmo è da subito visibilmente più efficiente e risultano corretti i calcoli.

invece, quando i = 5, alla iterazione successiva (la sesta):
a = 5*10^5
b = 1 * 10^20

notiamo invece che i due termini differiscono di un ordine di grandezza b + c inferiore alla precisione di macchina, quindi non si assiste ad una cancellazione e quindi: (a + b) + c = a + (b + c)

ottenendo effettivamente la seguente esecuzione:

```
(a+b)+c = 100000
a+(b+c) = 100000
```

questa invece è l'esecuzione per l'iterazione successiva(l'ultima) dove si può comunque apprezzare ciò riportato sopra:

```
(a+b)+c = 1e+006
a+(b+c) = 1e+006
```

ESERCIZIO 2:

2. Fissato l'intero positivo N, implementare un programma che permetta di calcolare $f_N(x)$ per il punto x e il grado N dati in input.
   Considerare i due algoritmi seguenti per i valori descritti dei parametri x e N, confrontando i risultati ottenuti per $f_N(x)$ con i valori restituiti per $f(x)$ dalla funzione exp della libreria ANSI *math.h*, tramite errore relativo e assoluto.
   - Algoritmo 1: determinare un'approssimazione di $f(x)$ per il punto $x = 0.5$ ed il punto $x = 30$, valutando $f_N(x)$ per $N = 3, 10, 50, 100, 150$. Ripetere l'esercizio considerando il punto $x = -0.5$ ed il punto $x = -30$.
   - Algoritmo 2: Osservando che per l'esponenziale vale $f(-x) = 1/f(x)$ e quindi $f(-x) \approx 1/f_N(x)$, determinare una diversa approssimazione di $f(-0.5)$ e $f(-30)$ nel modo seguente: valutare $f_N(+0.5)$ e $f_N(+30)$ per $N = 3, 10, 50, 100, 150$ e, successivamente, calcolarne il reciproco.

Svolgimento:

(in questo caso non viene riportato il codice data la maggiore concentrazione sull'esecuzione)

Analizziamo l'esecuzione dell'algoritmo 1(a sinistra) ed algoritmo 2(a destra):

```
Errore assoluto con x = 0.5 e N = 3: -0.00288794
Errore relativo con x=0.5 e N=3: -0.00175162
Errore assoluto con x = 0.5 e N = 10: -1.27627e-011
Errore relativo con x=0.5 e N=10: -7.74096e-012
Errore assoluto con x = 0.5 e N = 50: -4.44089e-016
Errore relativo con x=0.5 e N=50: -2.69354e-016
Errore assoluto con x = 0.5 e N = 100: -4.44089e-016
Errore relativo con x=0.5 e N=100: -2.69354e-016
Errore assoluto con x = 0.5 e N = 150: -4.44089e-016
Errore relativo con x=0.5 e N=150: -2.69354e-016

Errore assoluto con x = 30 e N = 3: -1.06865e+013
Errore relativo con x=30 e N=3: -1
Errore assoluto con x = 30 e N = 10: -1.06862e+013
Errore relativo con x=30 e N=10: -0.999978
Errore assoluto con x = 30 e N = 50: -3.18471e+009
Errore relativo con x=30 e N=50: -0.000298013
Errore assoluto con x = 30 e N = 100: 0.00390625
Errore relativo con x=30 e N=100: 3.65532e-016
Errore assoluto con x = 30 e N = 150: 0.00390625
Errore relativo con x=30 e N=150: 3.65532e-016

Errore assoluto con x = -0.5 e N = 3: -0.00236399
Errore relativo con x=-0.5 e N=3: -0.00389757
Errore assoluto con x = -0.5 e N = 10: 1.17416e-011
Errore relativo con x=-0.5 e N=10: 1.93586e-011
Errore assoluto con x = -0.5 e N = 50: -1.11022e-016
Errore relativo con x=-0.5 e N=50: -1.83045e-016
Errore assoluto con x = -0.5 e N = 100: -1.11022e-016
Errore relativo con x=-0.5 e N=100: -1.83045e-016
Errore assoluto con x = -0.5 e N = 150: -1.11022e-016
Errore relativo con x=-0.5 e N=150: -1.83045e-016

Errore assoluto con x = -30 e N = 3: -4079
Errore relativo con x=-30 e N=3: -4.35901e+016
Errore assoluto con x = -30 e N = 10: 1.21255e+008
Errore relativo con x=-30 e N=10: 1.29579e+021
Errore assoluto con x = -30 e N = 50: 8.78229e+008
Errore relativo con x=-30 e N=50: 9.38517e+021
Errore assoluto con x = -30 e N = 100: 6.71764e-005
Errore relativo con x=-30 e N=100: 7.17879e+008
Errore assoluto con x = -30 e N = 150: 6.71764e-005
Errore relativo con x=-30 e N=150: 7.17879e+008
```

```
Errore assoluto con x=-0.5 e N=3: 0.00106428
Errore relativo con x=-0.5 e N=3: 0.0017547
Errore assoluto con x=-0.5 e N=10: 4.69513e-012
Errore relativo con x=-0.5 e N=10: 7.74097e-012
Errore assoluto con x=-0.5 e N=50: 1.11022e-016
Errore relativo con x=-0.5 e N=50: 1.83045e-016
Errore assoluto con x=-0.5 e N=100: 1.11022e-016
Errore relativo con x=-0.5 e N=100: 1.83045e-016
Errore assoluto con x=-0.5 e N=150: 1.11022e-016
Errore relativo con x=-0.5 e N=150: 1.83045e-016

Errore assoluto con x=-30 e N=3: 0.000200763
Errore relativo con x=-30 e N=3: 2.14545e+009
Errore assoluto con x=-30 e N=10: 4.18699e-009
Errore relativo con x=-30 e N=10: 44744.2
Errore assoluto con x=-30 e N=50: 2.78952e-017
Errore relativo con x=-30 e N=50: 0.000298102
Errore assoluto con x=-30 e N=100: -3.78653e-029
Errore relativo con x=-30 e N=100: -4.04647e-016
Errore assoluto con x=-30 e N=150: -3.78653e-029
Errore relativo con x=-30 e N=150: -4.04647e-016
```

Analizziamo quindi i due algoritmi le quali esecuzioni sono riportate a sinistra quella dell'algoritmo 1 e sopra quella dell'algoritmo 2:

Algoritmo 1: come riportato in queste esecuzioni, calcola sia l'errore assoluto che l'errore relativo del calcolo dell'approssimazione di f(x) per i punti x = 0.5, 30, -0.5, -30 per N che ha valori 3, 10, 50, 100, 150.

Algoritmo 2: invece il secondo considera il reciproco come il risultato effettivo e viene effettuato solo per x = 0.5 e 30 e sempre con N che ha valori 3, 10, 50, 100, 150.
Questa volta quindi per ottenere l'approssimazione in quei punti come in precedenza, ma questa volta viene considerato il reciproco per trovare appunto direttamente quello dei punti x = -0.5 e -30 ottenendo rispetto all'algoritmo 1 dei risultati più precisi in alcune approssimazioni ottenendo errori relativi e assoluti minori ai precedenti.

ESERCIZIO 3:

3. Implementare un programma che determina la precisione di macchina eps, ossia il valore positivo $eps = 2^{-d}$, dove d è il più grande intero positivo tale che $1 + 2^{-d} > 1$ in aritmetica di macchina; calcolarne il valore sia in singola che in doppia precisione.

Svolgimento:

La precisione di macchina è apprezzabile grazie all'algoritmo implementato nell'esercizio 3:

Codice per il calcolo della precisione Singola:

```
void precisioneSingola()
{
    int d = 0;
    float f_eps, res;

    do
    {
        f_eps = pow (2, -d);
        res = 1 + f_eps;
        d++;
    } while(res > 1);

    cout << "eps con singola precisione: " << f_eps << endl << endl;
}
```

Esecuzione del codice sopra riportato:

```
eps con singola precisione: 5.96046e-008
```

Nel caso invece del calcolo della precisione Doppia. Il codice è il seguente:

```
void precisioneDoppia()
{
    int d = 0;
    double d_eps, res;

    do{
        d_eps = pow (2, -d);
        res = 1 + d_eps;
        d++;
    }while(res > 1);

    cout << "eps con doppia precisione: " << d_eps  << endl;
}
```

Mentre l'esecuzione:

```
eps con doppia precisione: 1.11022e-016
```