

Esercitazione 2.2 - spiegazione

Docente: Ing. Edoardo Fusella

Dipartimento di Ingegneria Elettrica e Tecnologie dell'Informazione

Via Claudio 21, 4° piano – laboratorio SECLAB

Università degli Studi di Napoli Federico II

e-mail: edoardo.fusella@unina.it

Esercitazione 2.2 - traccia

Programma che calcola la potenza

- L'utente inserisce un numero reale N e uno intero M , calcolare la potenza di N alla M con $M \geq 0$.
- Successivamente, abilitare il calcolo di N^M anche con $M < 0$.
- Modificare il programma in modo tale da chiedere all'utente dopo ogni calcolo se terminare l'esecuzione o continuare. Usare una variabile `char` che può assumere soltanto i valori 's' e 'n'.
- Infine, qualora l'utente inserisca una scelta differente, ripetere la domanda.

- Esempio:
N=2.3 M=3 Risultato: $N^M=12.167$
N=1.7 M=-4 Risultato: $N^M=0.11973$

Esercitazione 2.2 - algoritmo (1/2)

- Domanda: come calcolare N^M ?
- Risposta: moltiplicando N per se stesso M volte.
- Abbiamo bisogno di una variabile (chiamiamola *ris*) dove memorizzare i prodotti parziali.

Alla prima iterazione: $\text{ris}(\text{iterazione } 1) = N$

Alla seconda iterazione: $\text{ris}(\text{iterazione } 2) = N * N$

Alla terza iterazione: $\text{ris}(\text{iterazione } 3) = N * N * N$

Alla iterazione $M-1$: $\text{ris}(\text{iterazione } M-1) = N^{(M-1)}$

Alla iterazione M : $\text{ris}(\text{iterazione } M) = N^M$

Quindi

$\text{ris}(\text{iterazione } 1) = \text{ris}(\text{inizializzazione}) * N$

$\text{ris}(\text{iterazione } 2) = \text{ris}(\text{iterazione } 1) * N$

$\text{ris}(\text{iterazione } 3) = \text{ris}(\text{iterazione } 2) * N$

$\text{ris}(\text{iterazione } M) = \text{ris}(\text{iterazione } M-1) * N$

Esercitazione 2.2 - algoritmo (2/2)

- Esempio: come calcolare 3^4 ?
- Risposta: moltiplicando 3 per se stesso 4 volte.
- Abbiamo bisogno di una variabile (chiamiamola *ris*) dove memorizzare i prodotti parziali.

Alla prima iterazione: $\text{ris}(\text{iterazione } 1) = 3$

Alla seconda iterazione: $\text{ris}(\text{iterazione } 2) = 3 * 3$

Alla terza iterazione: $\text{ris}(\text{iterazione } 3) = 3 * 3 * 3$

Alla quarta iterazione : $\text{ris}(\text{iterazione } 4) = 3 * 3 * 3 * 3$

Quindi

$\text{ris}(\text{iterazione } 1) = \text{ris}(\text{inizializzazione}) * 3 = 1 * 3 = 3$

$\text{ris}(\text{iterazione } 2) = \text{ris}(\text{iterazione } 1) * 3 = 3 * 3 = 9$

$\text{ris}(\text{iterazione } 3) = \text{ris}(\text{iterazione } 2) * 3 = 9 * 3 = 27$

$\text{ris}(\text{iterazione } 4) = \text{ris}(\text{iterazione } 3) * 3 = 27 * 3 = 81$

Esercitazione 2.2 - codice

- la variabile ris viene usata per memorizzare sia i prodotti parziali che il risultato finale.
- Viene inizializzata ad 1 perché 1 è l'elemento neutro rispetto al prodotto.

```
1. #include<iostream>
2. #include<stdlib.h>
3. using namespace std;
4. int main(){
5.     float n = 0;
6.     int m = 0;
7.     float ris = 1;
8.     cout << "Inserire N:\n";
9.     cin >> n;
10.    cout << "Inserire M:\n";
11.    cin >> m;
12.    for(int i = 0; i<m; i++){
13.        ris *= n;
14.    }
15.    cout << "n^m = " << ris << " \n";
16.    system ("PAUSE");
17.    return 0;
18. }
```

Esercitazione 2.2 - codice

- Usiamo un for crescente come struttura di controllo iterativa.
- La variabile contatore di ciclo è di tipo intero e viene inizializzata a 0
- La condizione di terminazione è $i < m$
- Ad ogni iterazione i viene incrementata di 1
- Quindi:
 - alla prima iterazione, $i=0$
 - Alla seconda, $i=1$
 - Alla terza, $i=2$
 - Alla iterazione m , $i=m-1$
- Ci sono un totale di m iterazioni

```
1. #include<iostream>
2. #include<stdlib.h>
3. using namespace std;
4. int main(){
5.     float n = 0;
6.     int m = 0;
7.     float ris = 1;
8.     cout << "Inserire N:\n";
9.     cin >> n;
10.    cout << "Inserire M:\n";
11.    cin >> m;
12.    for(int i = 0; i<m; i++){
13.        ris *= n;
14.    }
15.    cout << "n^m = " << ris << " \n";
16.    system ("PAUSE");
17.    return 0;
18. }
```

Esercitazione 2.2 - codice

- Ad ogni iterazione si moltiplica il precedente risultato per n.
- Uguale a $\text{ris} = \text{ris} * n$;
- Esempio 3^4 :
 - Iterazione 1: $\text{ris} = 1 * 3 = 3$
 - Iterazione 2: $\text{ris} = 3 * 3 = 9$
 - Iterazione 3: $\text{ris} = 9 * 3 = 27$
 - Iterazione 4: $\text{ris} = 27 * 3 = 81$

```
1. #include<iostream>
2. #include<stdlib.h>
3. using namespace std;
4. int main(){
5.     float n = 0;
6.     int m = 0;
7.     float ris = 1;
8.     cout << "Inserire N:\n";
9.     cin >> n;
10.    cout << "Inserire M:\n";
11.    cin >> m;
12.    for(int i = 0; i<m; i++){
13.        ris *= n;
14.    }
15.    cout << "n^m = " << ris << " \n";
16.    system ("PAUSE");
17.    return 0;
18. }
```

Esercitazione 2.2 – M negativo

- Domanda: come calcolare N^M quando M è negativo?
- Risposta: usiamo la seguente uguaglianza $N^{-M} = \left(\frac{1}{N}\right)^M$
- Esempio: $2^{-3} = \left(\frac{1}{2}\right)^3$

Esercitazione 2.2 - codice

- Aggiungere le seguenti istruzioni prima del for

```
1. if (m<0){  
2.   n=1/n;  
3.   m=-m;  
4. }
```

```
1. #include<iostream>  
2. #include<stdlib.h>  
3. using namespace std;  
4. int main(){  
5.   float n = 0;  
6.   int m = 0;  
7.   float ris = 1;  
8.   cout << "Inserire N:\n";  
9.   cin >> n;  
10.  cout << "Inserire M:\n";  
11.  cin >> m;  
12.  for(int i = 0; i<m; i++){  
13.    ris *= n;  
14.  }  
15.  cout << "n^m = " << ris << " \n";  
16.  system ("PAUSE");  
17.  return 0;  
18. }
```

Esercitazione 2.2 - terminazione

- Domanda: Come gestire la terminazione del programma?
- Risposta:
 - Usiamo un ciclo while che contiene tutto il programma.
 - Quando la condizione del while è vera il programma continua.
 - Quando è falsa si esce dal while e il programma termina

Esercitazione 2.2 - codice

1. `char continuare = 's';`
2. `while(continuaire == 's'){`

- Dichiariamo una variabile di tipo `char` che si chiama *continuaire*
- Se *continuaire* è uguale a 's', il programma non termina

1. `cout<<"continuaire? (n=no) (s=si)\n";`
2. `cin>>continuaire;`
3. `}`

1. `#include<iostream>`
2. `#include<stdlib.h>`
3. `using namespace std;`
4. `int main(){`
5. `float n = 0;`
6. `int m = 0;`
7. `float ris = 1;`
8. `cout << "Inserire N:\n";`
9. `cin >> n;`
10. `cout << "Inserire M:\n";`
11. `cin >> m;`
12. `for(int i = 0; i<m; i++){`
13. `ris *= n;`
14. `}`
15. `cout << "n^m = " << ris << " \n";`
16. `system ("PAUSE");`
17. `return 0;`
18. `}`

Esercitazione 2.2 - terminazione

- Domanda: Come gestire la possibilità che l'utente inserisca un carattere diverso da 's' e 'n'?
- Risposta:
 - Usiamo un altro ciclo do..while:

```
1. do{  
2. cout<<"continuare? (n=no) (s=si)\n";  
3. cin>>continuare;  
4. }while(continuare!='s' && continuare!='n');
```

- Quando la condizione del do..while è vera (cioè finché continuare è diverso da 's' e 'n') il programma continua a ripetere la domanda.
- Quando è falsa (cioè quando continuare è uguale a 's' o 'n') si esce dal do..while e il programma continua