

(A) CONOSCENZA TERMINOLOGICA

Dare una breve descrizione dei termini introdotti:

- Passaggio parametri per valore
- Passaggio parametri per indirizzo
- Parametri formali e attuali
- Regole di visibilità
- Ambiente globale e locale
- Prototipo o firma
- Definizione di una funzione
- Dichiarazione di una funzione

(B) CONOSCENZA E COMPETENZA

Rispondere alle seguenti domande producendo anche qualche esempio

B1) Conoscenza

1. Cosa significa ambiente *globale* e *locale*?
2. Cosa rappresenta il *valore di ritorno* di una funzione?
3. Che differenza c'è tra *dichiarazione* di una funzione e *definizione* di una funzione?
4. A cosa serve la *visibilità delle variabili*?
5. Che tipo di *passaggio di parametri* consente Java?
6. Se una funzione riceve un parametro *passato per valore* e lo modifica, cosa si può affermare?
7. Come si può far restituire al programma chiamante, da parte di una funzione, un valore modificato?

B2) Competenza

1. Che relazione c'è tra parametri *formali* e *attuali*?
2. Perché conviene per quanto possibile usare *variabili locali*?
3. Quale è il vantaggio di scrivere *funzioni con parametri*?
4. Qual è la *sintassi dell'intestazione* di una funzione?
5. Come devono essere i *parametri formali* rispetto e quelli *attuali*?
6. Quando si usa la parola chiave **void**?

(C) ESERCIZI DI COMPrensIONE

1. Un programma formato da funzioni si dice e consiste nell'applicazione della tecnica Il problema principale risulta scomposto in che sono di difficoltà rispetto al problema principale.
2. Un sottoprogramma deve svolgere compito, essere di uso e rappresentare la soluzione di un singolo
3. Programmare mediante funzioni è vantaggioso perché:
 - a. il codice scritto è più
 - b. viene semplificata la dell'algoritmo;
 - c. risulta semplificato il durante l'esecuzione;
 - d. risulta semplificata la eventuale del codice.
4. Le variabili sono dichiarate all'esterno delle ed hanno visibilità nel programma, ossia possono essere e da tutte le funzioni.
5. Le variabili sono dichiarate all'interno delle singole ed hanno una visibilità limitata alla funzione in cui sono dichiarate. Nelle varie funzioni, si possono dichiarare variabili con lo stesso nome e modificarle senza rischio di confusione.
6. La parola chiave indica che il valore calcolato dalla funzione viene restituito al programma chiamante.
7. Per ciascuno degli esercizi seguenti, completare le parti mancanti, correggere eventuali errori, determinare gli output prodotti e dare una breve descrizione dell'applicazione

a. `int f1 (int x, int y)`
`{ float z;`
`z=(x>y)? x : y;`
`return z;`
`}`

b. `void f2 (int y)`
`{ x=pow(y,2);`
`}`

c. `int f3 (int x)`
`{`
`float a, b, c;`
`x=(x>y);`
`}`

e. `void f5 (int x, int y)`
`{`
`x=round (y,2);`
`}`

d. `char f4 (int x)`
`{`
`char y;`
`y=x++;`
`return y;`
`}`

f. `int f1 (int x, int y)`
`{`
`float z;`
`z=(x>y)? x : y;`
`return z;`
`}`

8. Data la funzione a fianco:
 - a. stabilire cosa calcola;
 - b. scrivere un opportuno programma **main()** che la istanzi;
 - c. proporre una versione alternativa equivalente.

```
int f(int x, int y)
{
  return (x<y)? (y-x) : (x-y);
}
```

9. La funzione a fianco dovrebbe scambiare il valore delle due variabili passate come parametro, ma contiene alcuni errori:
- individuare gli errori;
 - riscrivere la funzione corretta;
 - stabilire la corretta modalità di passaggio dei parametri;
 - scrivere un opportuno programma **main()** che la istanzi;

```
void SwapByReference (int v1, int v2)
{
    tmp=v1;
    v1=v2;
    v2=tmp;
}
```

10. La funzione a fianco deve calcolare la media di un numero di valori passato come parametro, ma contiene alcuni errori:
- individuare gli errori;
 - riscrivere la funzione corretta;
 - stabilire la corretta modalità di passaggio dei parametri;
 - scrivere un opportuno programma **main()** che la istanzi;

```
void calcolo(v)
{
    float s=0;
    int i, num;
    for (i=0; i<=v; i++)
    {
        cout<<"Inserisci il "<<i<<" numero: ";
        cin>>num;
        s=s+num;
    }
    return s;
}
```

11. La funzione a fianco deve calcolare il prodotto di x ed y mediante somme ripetute, ma contiene alcuni errori:
- individuare gli errori;
 - riscrivere la funzione corretta;
 - stabilire la corretta modalità di passaggio dei parametri;
 - scrivere un opportuno programma **main()** che la istanzi;
 - provando l'esecuzione con i valori 5 e -2 la funzione non dà risultati corretti. Modificarla in modo che operi anche con operandi negativi.

```
int prodotto(x, y)
{
    c=0;
    do
    {
        c=c+x;
        y- -;
    }
    while (y==0);
    return c;
}
```

(D) ESERCIZI DI APPLICAZIONE

Nella seguente serie di esercizi risolti, l'analisi è lasciata per esercizio al lettore; riportiamo la codifica in C++.

1. **Esercizio risolto.** Scrivere un programma che tramite una funzione calcoli il numero delle cifre di un dato valore intero, letto da input.

L'analisi è lasciata per esercizio; riportiamo

```
/* ContaCifre.cpp */
#include <iostream>
#include <cstdlib>
using namespace std;
int cifre (int n);
int main()
{
    int n;
    cout<<"Immetti il numero: ";
    cin>>n;
    cout<<"Il numero di cifre contenute in "<<n<<" e:"<<endl;
    cout<<cifre(n)<<endl;
    system("Pause");
    return EXIT_SUCCESS;
}
int cifre(int n)
{
    int c=0;
    do
    {
        n=n/10;
        c++;
    } while(n!=0);
    return(c);
}
```

2. **Esercizio risolto.** Scrivere un programma che, letto da input un intero n stabilisca se si tratti di un numero primo.

/* File: NumeroPrimo.cpp */

```
#include <iostream>
int primo(int x);
using namespace std;

int main()
{
```

```

int n;
cout<<"Immetti il numero: ";
cin>>n;
if (primo(n))
    cout<<n<<" e' primo";
else
    cout<<n<<" non e' primo";
cout<<endl;
system("Pause");
return EXIT_SUCCESS;
}

int primo(int x)
{
    int i;
    for(i=2;(i<=x/2) && (x%i);i++);
    if (i<=x/2)
        return(0);
    else
        return(1);
}

```

3. **Esercizio risolto.** Scrivere un programma che calcoli la potenza $Base^{Esponente}$ supponendo di non disporre della funzione `pow()`.

```

/* File: Potenza.cpp */
#include <iostream>
using namespace std;
int pot (int b, int e);
int main()
{
    int base, esponente;
    cout<<"Base: ";
    cin>>base;
    cout<<"Esponente: ";
    cin>>esponente;
    cout<<base<<" elevato a "<<esponente<<" e' "<<pot(base, esponente);
    system ("pause");
    return 0;
}

int pot(int b, int e)
{
    int i, p;
    i=1;
    p=1;
    while (i<=e)
    {
        p=p*b;
        i++;
    }
    return p;
}

```

4. **Esercizio risolto.** Scrivere un programma che consente l'immissione di una stringa come password e verifica che sia almeno di 6 caratteri, che contenga un uguale numero di caratteri alfabetici e numerici e che sia priva di simboli di punteggiatura.

```

/* Password.cpp */
#include <iostream>
#include <cstdlib>
#include <string.h>
#include <ctype.h>
using namespace std;
#define immissione "\nImmetti la password: "
#define reimmissione "\nPassword errata!"
#define corretta "\nPassword corretta!"
void lettura(string &p);
bool controllo1(string p);
bool controllo2(string p);
bool controllo3(string p);
int i;
int main()
{
    string risposta="",
        password;

```

```

do
{ lettura(password);
  if (controllo1(password) && /* controllo lunghezza minima */
      controllo2(password) && /* controllo assenza punteggiatura */
      controllo3(password)) /* controllo uguagl. car. alfabetici e numerici */
    risposta=corretta;
  else
    risposta=reimmissione;
  cout<<risposta<<endl;
} while (risposta!=corretta);
system("pause");
return 0;
}

void lettura(string &p)
{
  cout<<immissione;
  cin>>p;
}

bool controllo1(string p)
{
  if (p.size()<6)
    return false;
  else
    return true;
}

bool controllo2(string p)
{
  for(i=0; p[i] && !ispunct(p[i]);i++);
  if (ispunct(p[i]))
    return false;
  return true;
}

bool controllo3(string p)
{
  int a=0, d=0;
  for(i=0;p[i];i++)
  {
    if (isalpha(p[i])) a++;
    if (isdigit(p[i])) d++;
  }
  if(a!=d)
    return false;
  else
    return true;
}

```

5. **Esercizio risolto.** Un capitale c è conservato in banca per 3 anni. Il tasso è $t1$ per il primo anno, $t2$ per il secondo anno e $t3$ per il terzo anno. Supponendo che durante tre anni non avvengano prelievi, calcolare il capitale finale, allo scadere del terzo anno.

```

/* Capital.cpp */
#include <iostream>
#include <cstdlib>
using namespace std;
void lettura(float &c, float &t1, float &t2, float &t3);
float calcolo(float cap, float i);
void stampa(void);

int main()
{
  float c,t1,t2,t3;
  char risp;
  do
  {
    lettura(c, t1, t2, t3);
    cout<<"Ilvalore del montante è: "<<calcolo(calcolo(calcolo(c,t1),t2),t3)<<endl;
    cout<<"\n\nVuoi fare un altro calcolo ? (S/N)";
    cin>>risp;
  }
  while (toupper(risp)=='S');
  system("pause");
  return(0);
}

```

```

void lettura(float &c, float &t1, float &t2, float &t3)
{
    cout<<"Inserisci il valore del capitale iniziale:";
    cin>>c;
    cout<<"Inserisci il valore di t1:";
    cin>>t1;
    cout<<"Inserisci il valore di t2:";
    cin>>t2;
    cout<<"Inserisci il valore di t3:";
    cin>>t3;
}

```

```

float calcolo(float cap, float i)
{
    cap=cap+(cap/100*i);
    return(cap);
}

```

6. **Esercizio risolto.** Scrivere un programma che stampi le tavole aritmetiche progressive fino ad un dato valore *MAXN* letto da input.

```

/* Pwrtbl1.cpp */
#include <iostream>
#include <cstdlib>
using namespace std;
#define MAXN 10
int main()
{
    int i, sqr, cube, quad;
    cout<<" n\t n^2\t n^3\t n^4"<<endl; /* Intestazione colonne */
    cout<<"-----t-----t-----t-----"<<endl;
    for (i = 0; i<=MAXN; i++) /* Stampa la tabella dei valori */
    {
        sqr=i*i;
        cube=sqr*i;
        quad=cube*i;
        cout.width(4); /* ampiezza numero 4 cifre */
        cout.flags(ios::right); /* allineamento a destra */
        cout<<i<<"\t";
        cout.width(4);
        cout.flags(ios::right);
        cout<<sqr<<"\t";
        cout.width(4);
        cout.flags(ios::right);
        cout<<cube<<"\t";
        cout.width(5);
        cout.flags(ios::right);
        cout<<quad<<endl;
    }
    system("Pause");
    return (EXIT_SUCCESS);
}

```

7. **Esercizio risolto.** Sviluppare il precedente esercizio, facendo uso di funzioni.

```

/* Pwrtbl2cpp */
#include <iostream>
#include <cstdlib>
using namespace std;
#define MAXN 10
int sqr(int n);
int cube(int n);
int quad(int n);
int main()
{
    int i;
    cout<<" n\t n^2\t n^3\t n^4"<<endl; /* intestazione colonne */
    cout<<"-----t-----t-----t-----"<<endl;
    for (i = 0; i<=MAXN; i++) /* stampa righe */
    {
        cout.width(4);
        cout.flags(ios::right);
        cout<<i<<"\t";
        cout.width(4);
        cout.flags(ios::right);

```

```

    cout<<sqr(i)<<"\t";
    cout.width(4);
    cout.flags(ios::right);
    cout<<cube(i)<<"\t";
    cout.width(5);
    cout.flags(ios::right);
    cout<<quad(i)<<endl;
}
system("Pause");
return (EXIT_SUCCESS);
}

int sqr(int n)
{
    return n*n;
}
int cube (int n)
{
    return n*n*n;
}
int quad (int n)
{
    return n*n*n*n;
}

```

8. **Esercizio risolto.** Sviluppare il precedente esercizio, facendo uso di funzioni macro, esterne al file del programma. Si crea un file come il seguente, contenente le macro di interesse per il programma e posto nella stessa cartella del programma.

```

/* macros.h */
#define SQR(x) ((x) * (x))
#define CUBE(x) ((x) * (x) * (x))
#define QUAD(x) ((x) * (x) * (x) * (x))

```

Successivamente, si scrive il programma, che avrà un aspetto simile al seguente:

```

/* Pwrtb4.cpp */
#include <iostream>
#include <cstdlib>
#include "macros.h" /* riferimento al file di macro esterno */
using namespace std;
void StampaValore (int n, int prec); /* funzione di utilità per impostare una tantum */
#define MAXN 10 /* la precisione numerica */
#define colonne " \t n^2\t n^3\t n^4"
#define riga "-----\t-----\t-----\t-----"
int main()
{
    int i;
    cout<<colonne<<endl; /* Intestazione colonne */
    cout<<riga<<endl;
    for (i = 0; i<=MAXN; i++) /* Stampa la tabella dei valori */
    {
        StampaValore(i,4);
        StampaValore(SQR(i), 4);
        StampaValore(CUBE(i), 4);
        StampaValore(QUAD(i), 5);
        cout<<endl;
    }
    system("Pause");
    return (EXIT_SUCCESS);
}

void StampaValore (int n, int prec)
{
    cout.width(prec);
    cout.flags(ios::right);
    cout<<n<<"\t";
}

```

9. **Esercizio risolto.** Scrivere un programma che letti da input due interi D e d , calcoli il quoziente q ed il resto r della divisione, mediante un'unica funzione del tipo

```
void Dividi (int dividendo, int divisore, int &quoziente, int &resto)
```

in cui i parametri *quoziente* e *resto* sono passati per indirizzo.

```
/* PassaggioPer Indirizzo.cpp */
```

```

#include <iostream>
using namespace std;
void lettura(int &a, int &b);      /* lettura per indirizzo */
void Dividi(int dividendo,      /* calcolo divisione con passaggio */
            int divisore,       /* per indirizzo di quoziente e resto */
            int &quoziente, int &resto);

int main()
{
    int D, d, q, r;
    lettura(D, d);
    Dividi(D, d, q, r);
    cout<<"Quoziente: "<<q<<endl;
    cout<<"Resto: "<<r<<endl;
    system("Pause");
    return (0);
}

void Dividi(int dividendo, int divisore, int &quoziente, int &resto)
{
    quoziente=dividendo/divisore;
    resto=dividendo%divisore;
}

void lettura (int &a, int &b)
{
    cout<<"Dividendo: ";
    cin>>a;
    cout<<"Divisore: ";
    cin>>b;
}

```

10. Scrivere un'applicazione che consenta di risolvere un'equazione di secondo grado.
11. Scrivere una funzione per il calcolo dell'area di ciascuna delle seguenti figure geometriche:
 - a. quadrato
 - b. rettangolo
 - c. triangolo equilatero
 - d. triangolo isoscele
 - e. triangolo qualunque
 - f. trapezio rettangolo
 - g. trapezio scaleno
12. Scrivere un'applicazione per la risoluzione di un sistema di primo grado con due equazioni in due incognite.
13. Determinare la misura dell'ipotenusa di un triangolo rettangolo, note le misure dei cateti A e B, lette da terminale.
14. Dati 3 numeri in input che rappresentano l'orario di partenza di un aeroplano (H ore, M minuti e S secondi) e letto il tempo di permanenza in volo, espresso in secondi, determinare e stampare tramite una funzione l'orario di arrivo previsto.
15. Progettare un programma che consenta di stampare, tramite una funzione, la tavola pitagorica di ordine N, con N letto da input.
16. Dati tre numeri interi in ordine crescente, a, b e c, scrivere una funzione che stabilisca se possono essere le misure dei lati di un triangolo e, in caso affermativo, di che tipo.
17. Progettare una funzione che calcoli l'i-esimo numero della successione di Fibonacci, così definita:
$$\text{Fib}(0)=1; \text{Fib}(1)=1; \text{Fib}(i)=\text{Fib}(i-1) + \text{Fib}(i-2)$$

Mettere come esercizio risolto**Esempio 10**

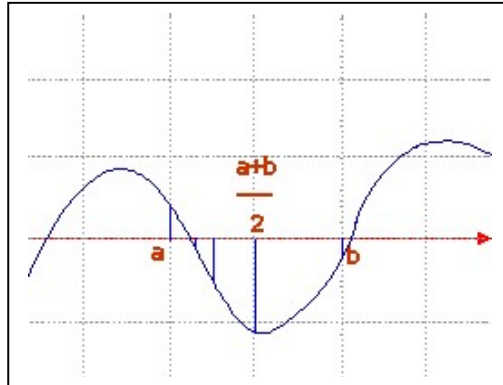
Si vuole realizzare un programma che calcoli in modo approssimato le soluzioni di un'equazione $f(x) = 0$ mediante il metodo della **bisezione**.

*Fase 1 - Analisi del problema**Analisi del testo*

Un metodo molto semplice per approssimare le soluzioni di un'equazione $f(x) = 0$ (o che è lo stesso di calcolare gli zeri della funzione $y = f(x)$ in un intervallo $[a, b]$) è quello della bisezione.

Prima di tutto è necessario separare le soluzioni e cioè individuare un intervallo $[x_1, x_2]$ nel quale vi sia almeno una soluzione, e cioè un intervallo nel quale ci sia un cambiamento di segno della funzione.

Nel disegno accanto la funzione $y=f(x)$ attraversa l'asse delle x tra x_1 e x_2 . Dunque vi è una soluzione tra x_1 e x_2 .



Per restringere il campo la cosa più semplice è

quella di dividere in due l'intervallo, calcolando la funzione nel suo punto medio. Dopodichè si individua in quale delle due metà si verifica ancora un cambiamento di segno: la soluzione si trova di certo in questo nuovo intervallo dimezzato. Il procedimento può ora ripetersi quante volte si vuole fino a ottenere la precisione voluta.

In pratica date x_1 e x_2 , si calcola il valore medio:

$$x = (x_1 + x_2) / 2$$

e quindi $y = f(x)$

Supponiamo che i 3 coefficienti siano numeri reali e che sia fissato un numero di cicli pari a 200; in alternativa questo valore potrebbe essere immesso da input. Supponiamo, inoltre, che la funzione di cui cercare gli zeri sia $y=x^2-5x$

Analisi delle specifiche di Ingresso/Uscita

Input	Nome	Tipo	Significato
	x1	Real	Primo estremo dell'intervallo
	x2	Real	Secondo estremo dell'intervallo
Output	msg	String	Messaggio all'utente
	x	Real	Soluzione dell'equazione

Tab. 0.7 Tabella delle variabili di I/O del problema di Esempio 10

Procedimento risolutivo

Si acquisiscono da input gli estremi dell'intervallo in cui cade almeno una soluzione di $f(x)$. Si calcolano le corrispondenti $y_1=f(x_1)$ e $y_2=f(x_2)$. Si esegue un ciclo di 2000 iterazioni in cui, ad

ogni iterazione, si effettua la seguente elaborazione: si calcola il punto medio x tra $x1$ ed $x2$, si calcola il corrispondente $y=f(x)$; se $y*y2>0$ si aggiorna $x2$ con x ed $y2$ con y , altrimenti si aggiorna $x1$ con x e $y1$ con y . Al termine del ciclo si stampa il valore di x , che fornisce una valutazione approssimata della soluzione di $f(x)$.

Fase 2 - Algoritmo

Algoritmo Bisezione

Inizio

Reale $x, x1, x2, y, y1, y2$;

Intero cicli, i ;

cicli=2000;

Leggi ($x1$);

Leggi ($x2$);

$y1=f(x1)$;

$y2=f(x2)$;

Per $i=1$ **a** cicli **fai**

Inizio

$x=(x1+x2)/2$;

$y=f(x)$;

Se ($y*y2>0$)

Inizio

$x2=x$;

$y2=y$;

Fine

Altrimenti

Inizio

$x1=x$;

$y1=y$;

Fine;

Fine; /* fine ciclo */

Fine.

Fase 3 – Codifica

/* Bisezione.cpp */

#include <iostream>

#include <stdlib.h>

#include <math.h>

using namespace std;

float f(float z);

10

```

int main(int argc, char *argv[])
{
    float x, x1, x2,
        y, y1, y2;
    int conta, cicli=2000000;
    cout<<"x1: ";
    cin>>x1;
    cout<<"x2: ";
    cin>>x2;
    y1=f(x1);
    y2=f(x2);
    for (conta=1; conta<cicli; conta++)
    {
        x=(x1+x2)/2;
        y=f(x);
        if (y*y2>0)
        {
            x2=x;
            y2=y;
        }
        else
        {
            x1=x;
            y1=y;
        }
    }
    cout<<"Soluzione: "<<x<<endl;
    system("PAUSE");
    return 0;
}

float f (float z)
{
    /* la funzione f consente di impostare una qualunque funzione
       matematica, algebrica o trascendente */
    return(z*z-5*z);
    // return (sin(z));
    // return (z*z-2);
    // return (z*z-5*z+6);
    // return (pow(z,3)-z+1);
    // return (pow(z,4)-pow(z,3)-7*pow(z,2)+z+6);
}

```