Ripasso procedure/funzioni e passaggio dei parametri

Esegui su un foglio i seguenti esercizi del file "Esercizi funz in C+problemi.pdf" (VEDI GLI ESEMPI SUCCESSIVI):

1 – 2 - 9 i primi 4 programmi – 17 – 18 – 22 – 23 – 24 – 27

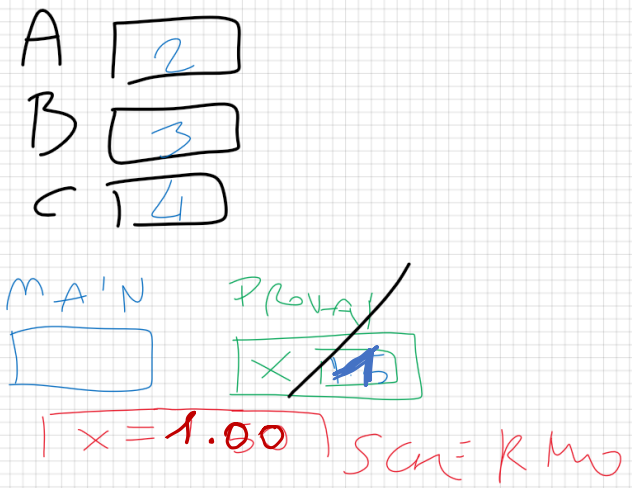
***Variabili globali**: nota che*

- *SCOPE: sono visibili in ogni parte del programma*
- *LIFETIME: occupano spazio per tutto il tempo di esecuzione del programma*

1. Dopo aver stabilito quali variabili so

```
int A,B,C;
void PROVA1(){
    float X;
    X=(A*B)/C;
    printf("X= %.2f",X);
    return;
}
int main(){

    A=2;
    B=3;
    C=4;
    PROVA1();
    return 0;
}
```

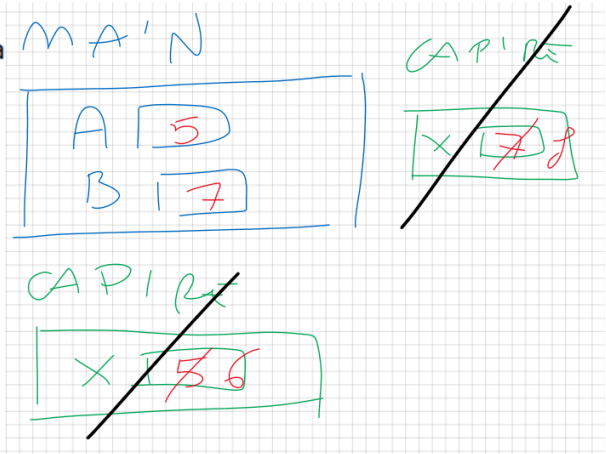


***Parametri passati per valore**: nota che*

- *i parametri formali sono come variabili locali, ma hanno già un valore iniziale = al valore del parametro attuale*
- *i parametri formali, come le variabili locali, occupano spazio per il solo tempo di esecuzione della procedura/funzione e sono visibili solo all'interno della procedura/funzione in cui sono stati dichiarati*
- *le modifiche effettuate al parametro formale NON si ripercuotono sul parametro attuale*

7. Dopo aver individuato le variabili globali e locali, i pa

```
void CAPIRE(int X){
    X=X+1;
    return;
}
int main(){
    int A,B;
    A=5; B=7;
    CAPIRE(A);
    CAPIRE(B);
    printf("A= %d B= %d",A,B);
    return 0;
}
```

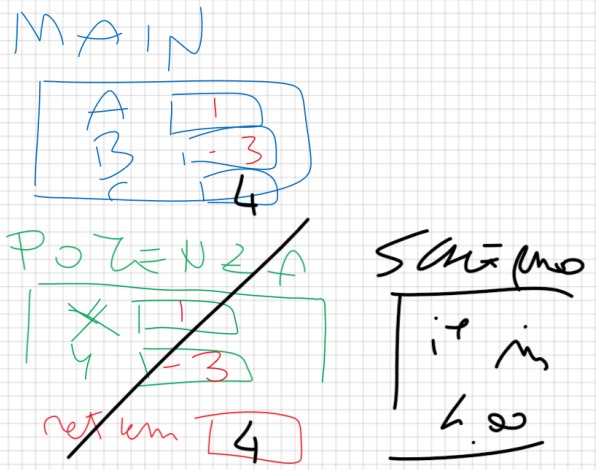


***Funzioni**: nota che*

- *restituiscono un valore dello stesso tipo della funzione*
- *questo valore sostituisce nel chiamante la sua chiamata*

16. Nel seguente programma determina variabili globali e l

```
float POTENZA(float X, float Y){
    return ((X+Y)*(X+Y));
}
int main(){
    float A,B,C;
    A=1;
    B=-3;
    C=POTENZA(A,B);
    printf("il risultato e' %.2f",C);
    return 0;
}
```

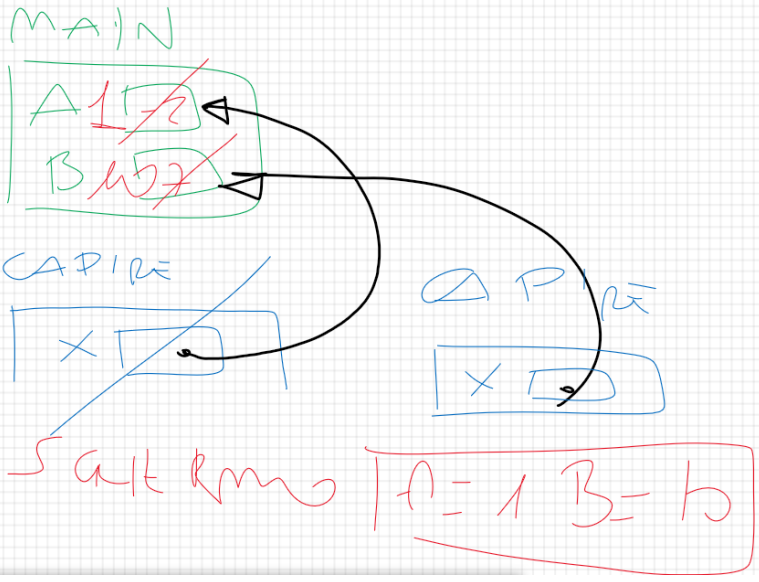


***Parametri passati per indirizzo**: nota che*

- *i parametri formali sono come variabili locali, ma hanno già un valore iniziale = all'INDIRIZZO del parametro attuale*
- *i parametri formali, come le variabili locali, occupano spazio per il solo tempo di esecuzione della procedura/funzione e sono visibili solo all'interno della procedura/funzione in cui sono stati dichiarati*
- *le modifiche effettuate al parametro formale SI ripercuotono sul parametro attuale*

21.

```
void Capire(int *x){
    *x = *x + 3;
    return;
}
int main(){
    int A,B;
    A=-2;
    B=7;
    Capire(&A);
    Capire(&B);
    printf("A= %d B= %d",A,B);
    return 0;
}
```



&A l'operatore & (**referenziazione**) restituisce l'indirizzo della cella in cui è memorizzata A

*x l'operatore * (**dereferenziazione**) restituisce il contenuto della cella il cui indirizzo è memorizzato in x

int *x; si legge "x è un puntatore a intero"

*x = *x + 3; si legge "alla cella puntata da x si assegna il valore della cella puntata da x + 3"