

I.I.S. "Marconi - Civitavecchia"

2C inf
2021-22

PROGRAMMA SVOLTO
SC. TECN. APPLICATE

Di Falco Mustazzella Vincenzo

DALL'ALGORITMO AL PROGRAMMA ESEGUIBILE

Algoritmi, diagrammi di flusso

Input e Output
Variabili e assegnamento
Scelta condizionata ed espressioni booleane
Sequenze di istruzioni

I **programmi**: scrittura del **codice sorgente**, compilazione, esecuzione del **codice eseguibile**

I **costrutti base** dei linguaggi di programmazione : Input, Output, assegnamento, scelta condizionata, cicli, sequenze e blocchi di istruzioni

Rappresentazione dei costrutti base nei **diagrammi di flusso**: I/O, assegnamento, scelta condizionata

Gli ambienti di sviluppo: definizione di **IDE**

Ambiente di sviluppo **Dev++** : creazione nuovo progetto, progetti GUI, console, **librerie** statiche e dinamiche, uso del compilatore ed esecuzione dei programmi
IDE on line **Repl** : uso del compilatore ed esecuzione dei programmi

Implementare un algoritmo espresso con diagramma di flusso:

Definizione di **istruzione** di input, di output
Definizione di **variabile**, l'istruzione di assegnamento
Definizione di **espressione**, valutazione di espressioni aritmetiche e booleane

Come la **RAM** e la **memoria di massa** vengono utilizzate durante l'esecuzione di un programma

Significato dei termini sintassi e semantica

Differenza tra linguaggi di **programmazione** e di **formattazione**, contesti di utilizzo.

Differenza tra compilatori e **interpreti**

Il supporto all'esecuzione (processore, S.O., librerie)

Linguaggio C

Struttura del programma C:

direttive include, librerie

funzione **Main** (tipo restituito, argomenti in input, corpo della funzione, istruzione return)

Istruzioni **cin** e **cout**, operatori pipe (>> e <<), librerie stdio e iostream, namespace std

Valori costanti: denotazione di interi, decimali, caratteri, stringhe, booleani

Espressioni costanti:

Operatori aritmetici (+, -, *, /, %)

Operatori boolean:

operatori di **confronto** (<, >, ==, !=)

operatori **logici** (AND, &&, OR, ||, NOT, !)

Il concetto di **variabile**, nome di variabile

variabili per l'input, l'output, per calcoli intermedi

concetto di **allocazione** di variabile in RAM, concetto di denotazione

concetto di assegnamento di un valore ad una variabile, concetto di valutazione di una variabile

Tipi di dato:

char, int, float, bool

Variabili:

Dichiarazione, **dichiarazione** con **inizializzazione**, modificatore **const**

Operatore e istruzione di **assegnamento** (=), operatore di uguaglianza VS assegnamento

Concetto di *scope* di una variabile, ciclo di vita di una variabile

Valutazione di variabile presente sia a sinistra che a destra dell'operatore di assegnamento

Operatori di **de/incremento** (-- e ++), differenza tra pre/postfisso

Tipi di dato:

short, long, double

funzione sizeof(), limiti numerici definiti in limits.h

Espressioni aritmetiche con variabili

Divisione intera e decimale

il **casting implicito** nella divisione tra variabili intere o/e decimali

libreria math.h, funzione sqrt(), il casting implicito e la funzione sqrt()

Casting esplicito

Tipo restituito e valutazione di espressioni aritmetiche (con variabili) composte

Istruzione **IF-ELSE**, condizione, blocco true e blocco false

Sintassi dell'istruzione IF-ELSE

IF-ELSE **annidati**

Diagrammi di flusso dei blocchi IF-ELSE annidati

Espressioni booleane con variabili

Scrittura e valutazione di espressioni booleane (con variabili) composte

Array:

descrizione

Dichiarazione e inizializzazione

indici delle celle

R/W di singoli elementi: accesso casuale ad una cella tramite indice costante, tramite indice espresso da variabile

Ciclo FOR :

Introduzione, utilità d'uso

La sintassi del FOR : controlli (inizializzazione, condizione, update), corpo

Blocco per il ciclo FOR nei diagrammi di flusso

Quando si esce dal FOR?

Ciclo FOR e vettori:

variabili indice (inizializzazione, condizione, incremento)

visita sequenziale diretta e inversa delle celle di un vettore

visita sequenziale con input negli elementi di un vettore

visita sequenziale con output degli elementi di un vettore

visita sequenziale e selezione degli elementi di un vettore in base a condizioni sul valore o sull'indice (con operazioni di R/W)

Funzioni:

Descrizione introduttiva, vantaggi d'uso, funzioni VS procedure

Ruolo della dichiarazione, definizione, invocazione di una funzione, il passaggio dei parametri

Esempio di invocazione: la funzione rand()

Sintassi di **dichiarazione** di funzione, parametri **formali**, tipo restituito

Sintassi della **definizione** di funzione, dichiarazione di variabili **locali** alla funzione,

scope dei parametri attuali e delle variabili locali, l'istruzione **return**

Sintassi della **invocazione** di funzione, parametri **attuali**, casting implicito nel passaggio di parametri

Valutazione dell'invocazione di funzione in istruzioni di assegnamento e nelle espressioni

Il ciclo di vita di parametri attuali e variabili locali

ciclo **WHILE** :

Introduzione, condizione e blocco true, quando preferirlo al FOR

sintassi dell'istruzione WHILE

Blocco per il ciclo WHILE nei diagrammi di flusso

Quando si esce dal WHILE?

Alcuni esempi utili trattati

sequenze:

scambio di valore tra due variabili (con e senza variabile appoggio)

scelta condizionata:

confronto di due date

variabili che contano e accumulano:

quanti zeri, moltiplicazione con la somma, elevamento a potenza con la moltiplicazione

cicli di selezione:

trova il numero

trova massimi e minimi

azzerà i valori pari in un array

manda in output gli elementi con indice dispari

cicli con accumulo:

quanti dispari

media

conversione binario-decimale

funzioni:

area triangolo e cerchio

while:

controllo password

pari o dispari?