**23.12.21 - Algoritmi - recupero e potenziamento primo trimestre**

Scade il 11 gennaio 2024 23:59

**Istruzioni**

Eseguire l'analisi di seguenti problemi, dettagliando dati in input/output/lavoro e, successivamente, creare gli algoritmi risolutivi:

1. Un amministratore di condominio deve calcolare la quota mensile che il singolo condomino deve pagare per la manutenzione degli spazi verdi comuni. Tale **quota** è data dalla **somma della quota base e della quota accessoria**, che vale 5€ per ogni mq dell'abitazione del condomino. Infine sulla quota complessiva (data dalla somma della quota base e della quota accessoria) il condomino ottiene una **agevolazione** (cioè uno sconto) pari a **10€ per ogni figlio**. Calcolare la quota che il condomino deve pagare presi in **input** la **quota base**, i **mq** **dell'appartamento** e il **numero di figli**.

Input: qbase (quota base), superfice (mq appartamento), nfigli (numero di figli)

Output: qtotale (quota che il condomino deve pagare)

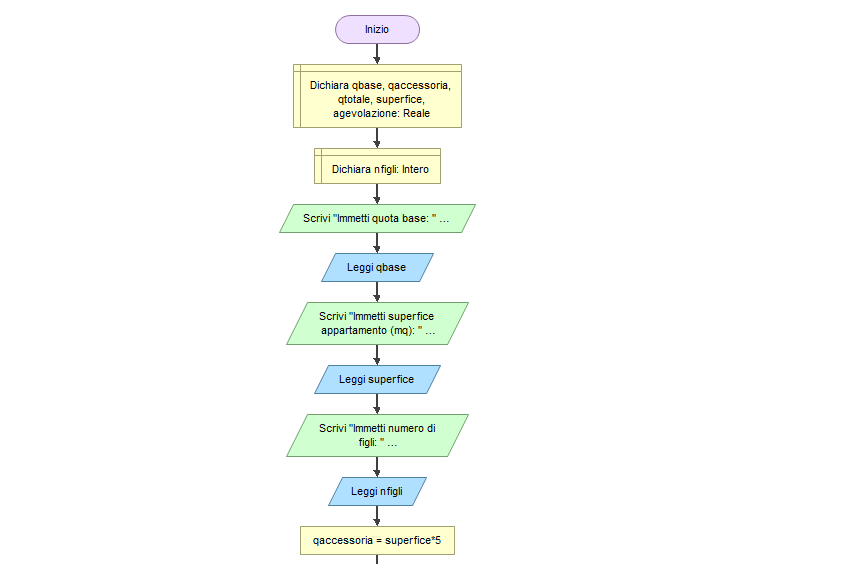
Lavoro:

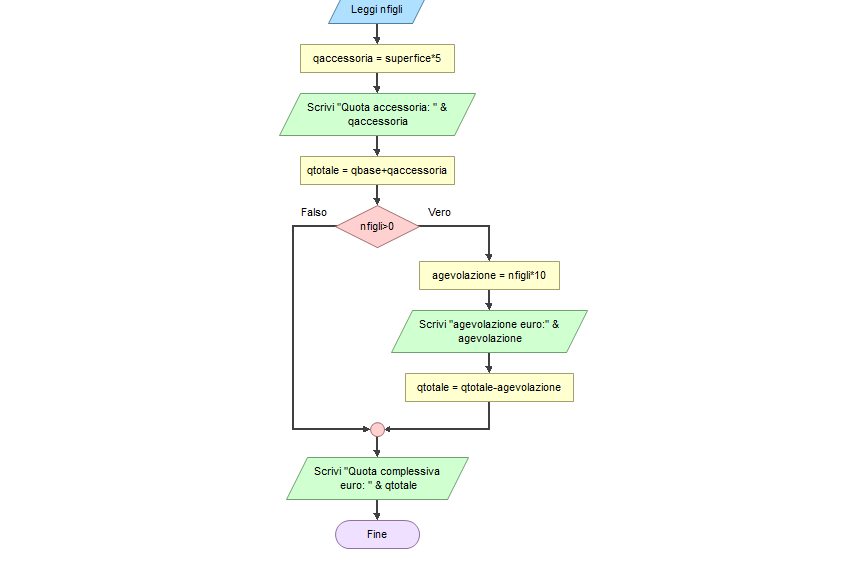
qaccessoria (quota accessoria) = superfice \* 5,00 €

agevolazione = nfigli \* 10,00 €

qtotale = qbase - qaccessoria

Algoritmo





Codifica in C++

#include <iostream>

using namespace std;

int main() {

double qbase, qaccessoria, qtotale, superfice, agevolazione;

int nfigli;

cout << "Immetti quota base (euro): ";

cin >> qbase;

cout << "Immetti superfice appartamento (mq): ";

cin >> superfice;

cout << "Immetti numero di figli: ";

cin >> nfigli;

qaccessoria = superfice \* 5;

cout << "Quota accessoria: " << qaccessoria << endl;

qtotale = qbase + qaccessoria;

if (nfigli > 0) {

agevolazione = nfigli \* 10;

cout << "Agevolazione (euro):" << agevolazione << endl;

qtotale = qtotale - agevolazione;

}

cout << "Quota complessiva (euro): " << qtotale << endl;

return 0;

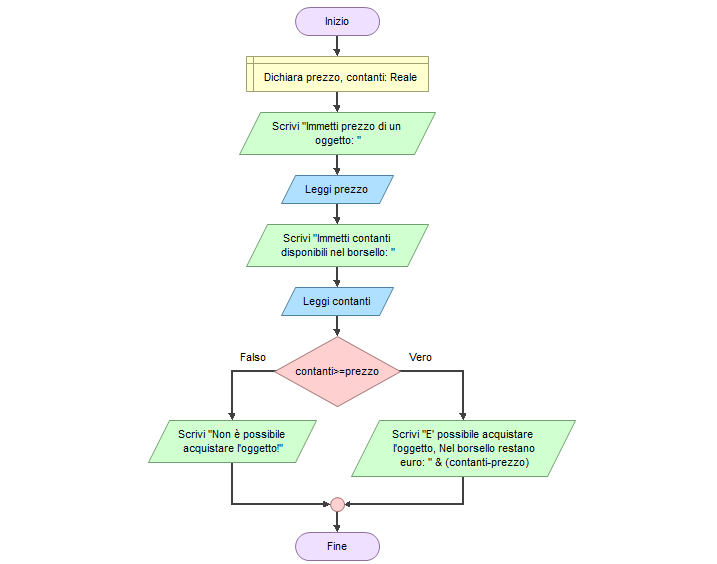
}

1. Conoscendo il prezzo di un oggetto e la cifra a disposizione nel borsello, determinare se è possibile acquistare o non acquistare l’oggetto; se è possibile verificare se, e quanti, soldi rimangono nel borsello dopo l’acquisto.

Input: prezzo (prezzo oggetto), contanti (cifra disponibile nel borsello)

Output: se possibile acquistare l’oggetto ed eventuale rimanenza contanti - prezzo

Algoritmo



Codifica in C++

#include <iostream>

using namespace std;

int main() {

double prezzo, contanti;

cout << "Immetti prezzo di un oggetto: ";

cin >> prezzo;

cout << "Immetti contanti disponibili nel borsello: ";

cin >> contanti;

if (contanti >= prezzo) {

cout << "E' possibile acquistare l'oggetto. Nel borsello restano euro: " << contanti - prezzo << endl;

}

else {

cout << "Non e' possibile acquistare l'oggetto!" << endl;

}

return 0;

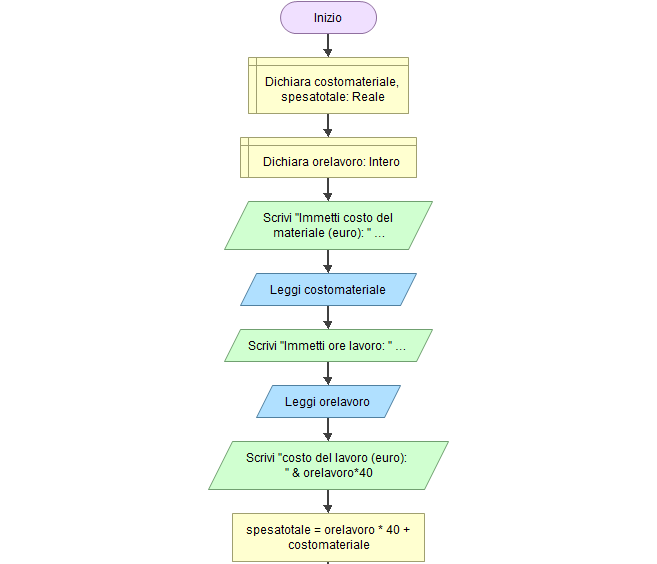
}

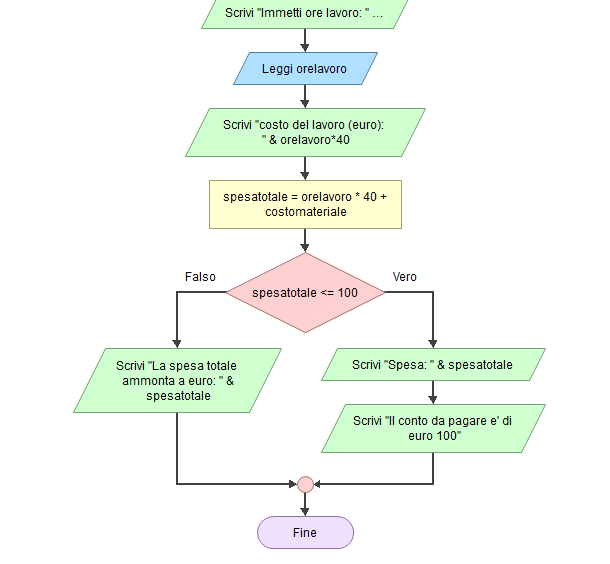
1. Un idraulico chiede 40,00€ per un’ora di lavoro, più il **costo del materiale**, con un minimo di 100,00€ per ogni lavoro. Dato il **costo del materiale** e il **numero di ore lavorative**, calcolare la **spesa totale**, facendola ammontare a 100,00€ dove occorre.

Input: costomateriale, orelavoro

Output: spesatotale = orelavoro \* 40 + costomateriale

Algoritmo





Codifica in C++

#include <iostream>

using namespace std;

int main() {

double costomateriale, spesatotale;

int orelavoro;

cout << "Immetti costo del materiale (euro): ";

cin >> costomateriale;

cout << "Immetti ore lavoro: ";

cin >> orelavoro;

cout << "costo del lavoro (euro): " << orelavoro \* 40 << endl;

spesatotale = orelavoro \* 40 + costomateriale;

if (spesatotale <= 100) {

cout << "Spesa: " << spesatotale << endl;

cout << "Il conto da pagare e' di euro 100" << endl;

}

else {

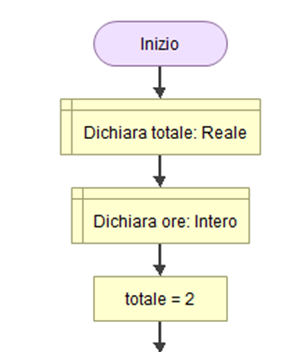
cout << "La spesa totale ammonta a euro: " << spesatotale << endl;

}

return 0;

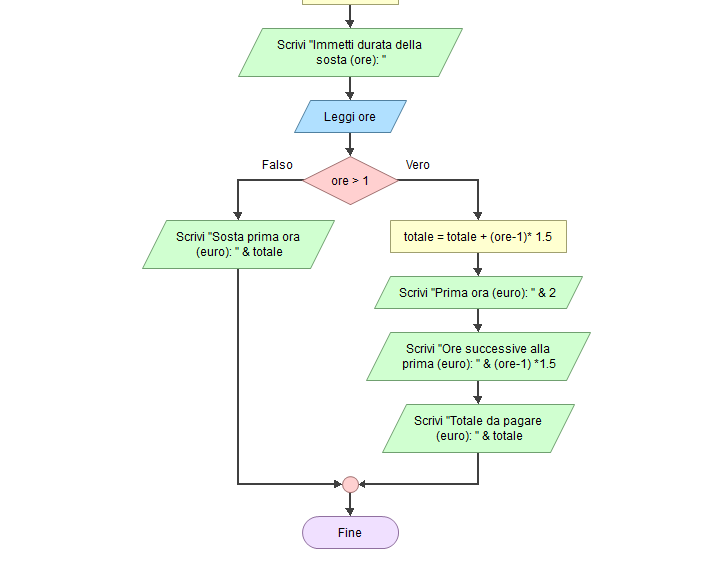
}

1. Sapendo che in un parcheggio la prima ora costa €2 mentre tutte le successive costano €1,5, scrivere un programma che richieda il numero delle ore e visualizzi il totale da pagare.

Input: ore

Output: totale

Algoritmo



Codifica in C++

#include <iostream>

using namespace std;

int main() {

double totale=2;

int ore;

cout << "Immetti durata della sosta (ore): " << endl;

cin >> ore;

if (ore > 1) {

totale = totale + (ore - 1) \* 1.5;

cout << "Prima ora (euro): " << 2 << endl;

cout << "Ore successive alla prima (euro): " << (ore - 1) \* 1.5 << endl;

cout << "Totale da pagare (euro): " << totale << endl;

}

else {

cout << "Sosta prima ora (euro): " << totale << endl;

}

return 0;

}

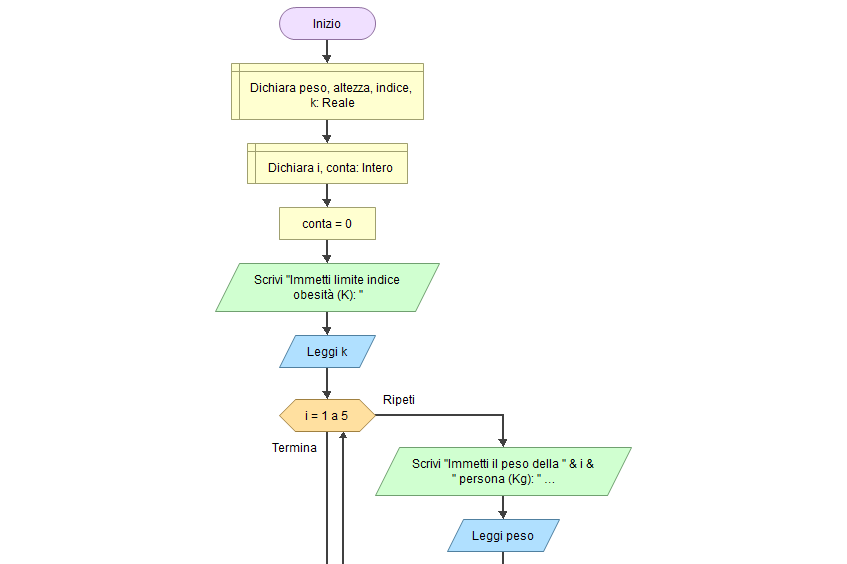
1. Sono dati il peso e l'altezza di 5 persone. Calcolare per ognuna l'indice di obesità = peso diviso l'altezza. Si conti man mano il numero di quelle che hanno obesità maggiore di un valore K prefissato.

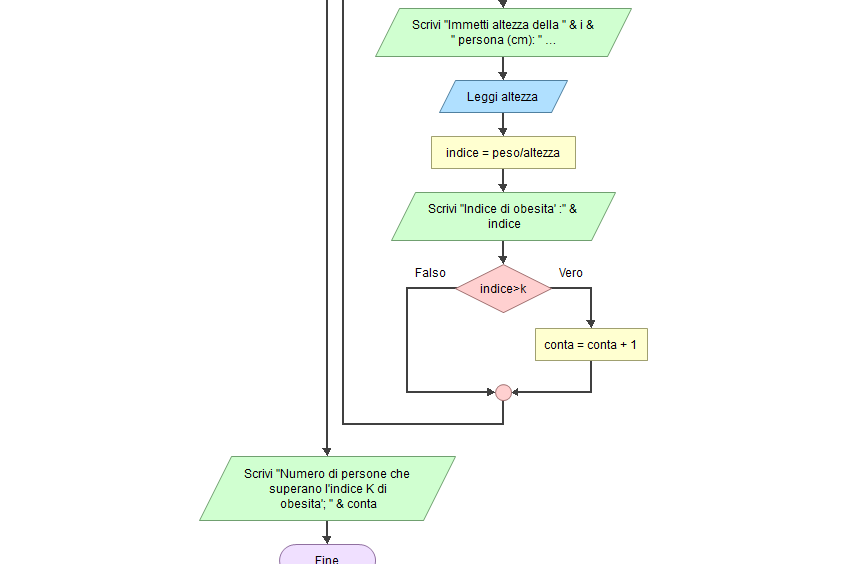
Input: k (indice limite di obesità), peso e altezza (di ciascuna delle 5 persone)

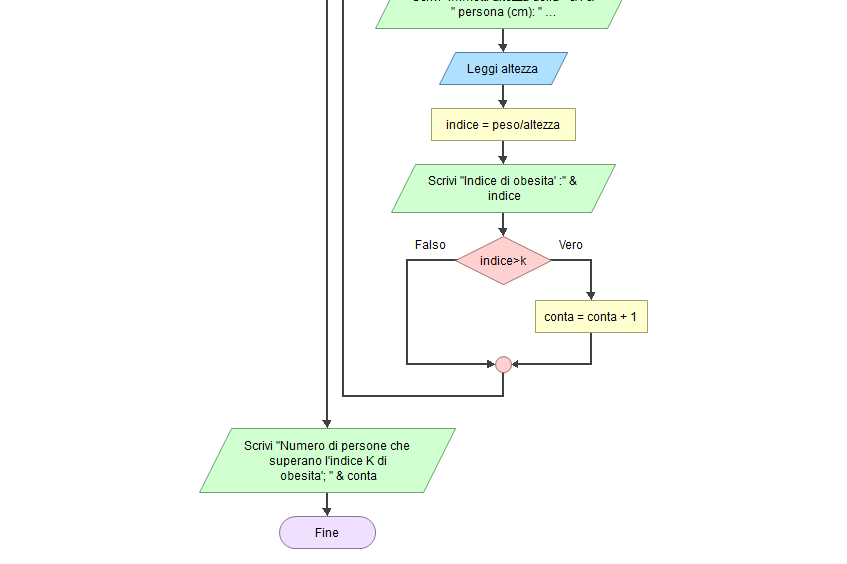
Output: indice (di obesità di ciascuna delle 5 persone), conta (numero delle persone con obesità maggiore di K)

Lavoro: i (indice del ciclo for da 1 a 5)

Algoritmo







Codifica in C++

#include <iostream>

using namespace std;

int main() {

double peso, altezza, indice, k;

int i, conta;

conta = 0;

cout << "Immetti limite indice obesita' (K): ";

cin >> k;

for (i = 1; i <= 5; i++) {

cout << "Immetti il peso della " << i << " persona (Kg): ";

cin >> peso;

cout << "Immetti altezza della " << i << " persona (cm): ";

cin >> altezza;

indice = peso / altezza;

cout << "Indice di obesita' :" << indice << endl;

if (indice > k) {

conta = conta + 1;

}

}

cout << "Numero di persone che superano l'indice di obesita' (K="<< k <<"): " << conta << endl;

return 0;

}