



ISTITUTO ISTRUZIONE SUPERIORE "A. AVOGADRO"

(ENTE DOTATO DI PERSONALITA' GIURIDICA E DI AUTONOMIA AMMINISTRATIVA, ORAGNIZZATIVA e DIDATTICA - R. D. 24.08.1933 N. 21933 - DP. N. 3917-P/C16 DEL 7.3.2000)
C.A.P. 10124 - TORINO - Corso S. Maurizio, 8 - Tel. 011 81.53.611 fax 011 81.53.750 Cod. Mec. TOIS05100C
E-Mail Certificata: tois05100c@pec.istruzione.it E-Mail ordinaria: tois05100c@istruzione.it



INFORMATICA

Programma svolto nell'a.s. 2018/2019

Classe 3AL (Liceo Scientifico – opzione scienze applicate)

Docente: prof. Daniele Steindler

LINGUAGGIO C++

Nel dettaglio

- La struttura di un programma
- Gli statement
- La dichiarazione delle variabili
- Le costanti tipizzate e le definizioni per il preprocessore
- L'inizializzazione
- Il concetto di tipo di dato
- I tipi di dato fondamentali (int, float, char, bool,
- Introduzione alle stringhe
- Le frasi di commento
- L'assegnazione dei valori alle variabili
- Gli operatori aritmetici +, -, *, /, %
- Le forme contratte +=, -=, *=, /=, %=, >>=, <<=, &=, ^=, |=
- Incremento and decremento (++ , --)
- La divisione intera e il resto
- Il casting esplicito
- Gli operatori di relazione (==, !=, >, <, >=, <=) e logici (!, &&, ||)
- La precedenza tra gli operatori
- Le istruzioni di input e output (cin, cout, getline)
- Le fasi della programmazione
- La struttura di sequenza
- La struttura di selezione
 - If
 - If else
 - If annidato
 - Il problema del dandling else
 - If else if else if
 - Switch
- L' iterazione
 - While
 - Do while
 - For
 - Break

- Continue
- La generazione dei numeri casuali
- Cicli annidati (problemi dei numeri amichevoli, fidanzati, deficienti, perfetti, ricerca dei numeri perfetti tra 1 e 10000, eccetera)
- Condizioni complesse
- Le funzioni della libreria cmath e l'applicazione a problemi di altre discipline scientifiche
- La libreria ctype
- La serie di Fibonacci
- La serie buffa e altre successioni numeriche
- Le funzioni
- Stack di chiamate
- La definizione di una funzione
- Il richiamo
- Il prototipo
- Parametri formali
- Parametri attuali
- La restituzione del valore di ritorno
- Il passaggio dei parametri per valore
- Il passaggio dei parametri per riferimento
- Lo "scope" o spazio di visibilità delle variabili
- Le variabili globali
- Gestione in memoria delle chiamate delle funzioni e della restituzione del valore.
- I vettori.
- Inizializzazione.
- Accesso ai valori.
- Gli array di valori numerici.
- Stampa.
- Stampa a rovescio.
- Ribaltamento.
- Ricerca di valori.
- Esercizi sui vettori (pag. 60 del testo e scheda scaricabile dal link al fondo di pag 61 su libro in ed. digitale).

Compiti per le vacanze

1. Completare

<pre> #include <iostream> using namespace std; main() { int giorno; cout << "inserisci giorno: "; cin >> giorno; switch(giorno) { case 1: cout << "Lunedì\ " << endl; case 2: cout << "Martedì\ " << endl; case 7: cout << "Domenica " << endl; } } </pre>	
--	--

<pre> } system("PAUSE"); } </pre>	
-----------------------------------	--

2. Comprendere e spiegare il funzionamento dei seguenti programmi C++

<pre> #include <iostream> using namespace std; main() { int a,b,c; float media; //e se metto int cosa succede? cout << "Inserisci il primo voto: "; cin >> a; cout << "Inserisci il secondo voto: "; cin >> b; cout << "Inserisci il terzo voto: "; cin >> c; if((a>=18) && (a<=30) && (b>=18) && (b<=30) && (c>=18) && (c<=30)) { media=(a+b+c)/3.0; cout << "Media dedi valori inseriti=" << media << endl ; } system("PAUSE"); } </pre>	
<pre> #include <iostream> using namespace std; main() { int a=443, b=19; float c; c = (float)a/b; //prova togliendo (float) cout << "La media e\' " << c << endl; system("PAUSE"); } </pre>	
<pre> #include <iostream> using namespace std; main() { int mese,anno; int giorniMese; cout << "Inserisci il mese(1..12) e anno "; cin >> mese >> anno; switch (mese) { case 1: case 3: case 5: case 7: case 8: case 10: case 12: giorniMese = 31; break; case 4: case 6: case 9: case 11: giorniMese = 30; break; case 2: if(anno % 4 == 0 && anno != 1900) giorniMese = 29; else giorniMese = 28; break; } cout << "il mese in questione ha giorni " << giorniMese << endl; system("PAUSE"); } </pre>	
<pre> #include <iostream> using namespace std; main() { int numero, somma; cout << "Inserire un numero intero positivo (0 per terminare) "; cin >> numero; for (somma = 0; numero;) { somma += numero; cout << "Inserire un numero intero positivo (0 per terminare) "; cin >> numero; } cout << "La somma e\' " << somma << endl; system("PAUSE"); } </pre>	

--	--

3. Esercizi di riepilogo

- a. Scrivi un programma C++ che legga N numeri interi e calcoli la somma dei soli numeri dispari tra questi
- b. Scrivi un programma che legga da tastiera N numeri (con N fornito in input) e restituisca in output il valore minimo e il valore massimo
- c. Scrivi un programma C++ che visualizzi i primi N numeri primi
- d. Scrivi un programma in C++ che, servendosi esclusivamente delle operazioni di addizione e sottrazione calcoli rispettivamente il prodotto e il quoziente di due numeri interi positivi X e Y (immessi da tastiera). Per quanto riguarda il quoziente, accertati di non incorrere in forme indeterminate.
- e. Supponendo che l'esecutore conosca le quattro operazioni, scrivi un programma in C++ per il calcolo della potenza X^Y di due interi non negativi X e Y ricevuti in ingresso. Riscrivi poi il programma in C++ nel caso in cui l'esecutore conosca solo le operazioni di somma e sottrazione.
- f. Si dispone di un elaboratore capace di calcolare solamente il precedente ed il successivo di un numero. Scrivi per questa macchina in C++ un programma che calcoli la somma di due numeri X e Y. (Suggerimento: la somma richiesta può essere ottenuta aggiungendo a X tante unità quante se ne possono togliere a Y.)

4. Esercizi sulle funzioni:

- a. Senza parametri:
 1. Scrivere un programma in C++ che contenga una funzione che stampa la tavola pitagorica
 2. Scrivere un programma C++ che a scelta dell'utente richiami una tra tre funzioni che rispettivamente visualizzano i numeri da 1 a 20, i primi 20 numeri pari, i primi 20 numeri dispari
- b. Con parametri **NEI SEGUENTI ESERCIZI E' OBBLIGATORIO UTILIZZARE LE FUNZIONI CON PARAMETRI:**
 1. Scrivere un programma C++ composto da una funzione che riceva un numero intero e visualizzi in output tanti asterischi quanti sono indicati dal parametro trasmesso. Il numero intero non può essere maggiore di 10.
 2. Scrivere un programma C++ che, a scelta dell'utente realizzi la moltiplicazione o la divisione di due numeri interi positivi, servendosi solo dell'operazione di addizione e sottrazione
 3. Scrivere un programma C++ che, data in input una sequenza di numeri chiusa dallo zero, determini tutti i numeri che risultano maggiori della somma di tutti i precedenti
 4. Scrivere un programma in C++ con una funzione che restituisca nella variabile passata come primo parametro il suo valore moltiplicato per il valore del secondo. Cioè alla funzione occorre passare una variabile intera A ed un secondo parametro X. In uscita la variabile passata come primo parametro dovrà valere $A*X$.
 5. Scrivere un programma C che contenga una funzione che riceva in input la base e l'altezza di un rettangolo e restituisca come valore del terzo e del quarto parametro perimetro e area

5. Esercizi Array

- a) Scrivere un programma in C++ che, caricato un array di 10 numeri stampi prima tutti gli elementi in posizione pari e poi tutti quelli in posizione dispari. Inoltre calcoli la somma di tutti gli elementi in posizione pari e di tutti gli elementi in posizione dispari.
- b) Scrivere un programma in C++ che, caricato un array di 10 numeri, interi calcoli:
1. La somma e la media di tutti i numeri pari;
 2. La somma e la media di tutti i numeri dispari;
 3. La somma e la media di tutti i numeri positivi;
 4. La somma e la media di tutti i numeri negativi;
 5. La media di tutti i 10 numeri;
- Si produca inoltre la stampa della differenza tra ciascun numero e la media generale. Quanto vale la somma di queste differenze? (calcola e stampa)
- c) compilazione di un array (chiamato 'arpot') con n elementi (n in input) $1^1, 2^2, \dots, n^n$.
- d) dati min e max contare quanti elementi di 'arpot' sono compresi tra min e max
- e) compilare un array (chiamato 'arcas') con numeri casuali compresi tra min e max forniti in input
- f) contare dato un numero n quanto è lunga la sequenza più lunga di ripetizioni di n in 'arcas'
- g) contare quanto è lunga la sequenza più lunga di ripetizioni consecutive di un numero qualunque in arcas
- h) verificare se l'array arcas è ordinato in modo crescente o decrescente
- i) verificare se l'array arcas ha al suo interno una sequenza crescente o decrescente più lunga di un numero n inserito in input
- l) trovare la sequenza crescente più lunga in arcas e comunicare quanto è lunga e i suoi estremi