**INTRODUZIONE:**

**Termini da usare sullo pseudocodice:**Se, Allora, Altrimenti, Fine, Mentre, Ripeti, Finché.  
**Come mandare i file al prof:**  
Tipo: Pdf o Txt .  
Nome del file : CognomeNome (Es:AbbattistaFabio.Pdf).  
**Strutturazione esercizi:**  
1: Parte di input:  
Quello che mi serve avere  
2: Parte di output:  
Quello che ottengo eseguendo l’algoritmo.  
3: Parte di Pseudocodice:  
Tutte le operazioni da svolgere e la seguenza con cui svolgerle.  
**Annotazioni:**  
Problem solving: L’algoritmo rappresenta la proposta risolutiva al problema.  
Tipi di algoritmi: Pseudocodice, diagramma di flusso e linguaggio naturale.  
Una volta aver fatto l’algoritmo di pseudocodice bisogna verificarlo.  
**Verifica:**  
- Correttezza: se funziona in 1 caso non vuol dire funzioni in altri casi.  
- Efficienza, solo e soltanto operazioni essenziali per risolvere il problema meno operazioni   
 inutili.  
- Leggibilità.  
**Passi risolutivi dell’algoritmo:**  
Capire di che problema si tratta   
Analisi del problema,   
Dati di partenza(cosa rappresenta, tipo di valori e vincoli)   
Risultato.  
**Caratteristiche algoritmo:**  
-Finitezza  
-Non ambigua  
-Comprensibilità  
 **Es algoritmo:**  
**Somma di 2 numeri interi positivi:**5+7=12.  
Per farlo facciamo: 5+1, 6+1, 7+1, 8+1.  
**Prodotto di 2 numeri interi positivi con la strutturazione del prof:  
Dati input:**   
A: 1 fattore del prodotto, Num nat > 0  
B: 2 fattore del prodotto, Num nat > 0  
**Dati output:**   
C: Prodotto da A e B, Num nat >0  
Algoritmo del prodotto in italiano: Sommare ad A se stesso un numero di volte pari al valore di B (è ambiguo allora usiamo lo pseudocodice)  
**Pseudocodice:**  
Mentre (B>0)  
 C=C+A  
 B=B-1  
Fine  
Risultato=C **Oppure**Inizio  
Prodotto=0  
Mentre(B>0)  
Altrodato= A  
Mentre (Altrodato>0)  
Prodotto=Prodotto+1  
Altrodato = Altrodato -1  
Fine  
B=B-1  
Fine



**Unione di insiemi algoritmo:**AB = {A+B} – {AB}

**Input/output:**  
L1= Primo insieme.  
L2= Secondo insieme.  
Ls= Somma L1 e L2  
Li= intersezione L1 e L2  
Lu= Ls-Li  
**Pseudo codice:**  
Ls: Sommare Insiemi (L1, L2)  
Li: intersezione Insiemi (L1, L2)  
Lu: Sottrarre Insiemi (Ls, Li)  
Composizione di funzioni  
y= f(x); z= g(y); z= g(f(x));  
Si può scrivere cosi:  
AB = {A+B} – {AB}  
Lu = Sottrarre Insiemi(Sommare insieme(I1,I2), Intersezione Insiemi (I1,I2))

Dati 2 elenchi di interi A e B ordinare l’elenco A in maniera crescente, inerire l’elenco B in A mantenendo l’ordine crescente.  
Se abbiamo già per esempio l’ordinazione dell’elenco, abbiamo già del lavoro fatto .

Invece di usare elenco ora usiamo array.  
Elementari nel c:  
Reali, interi, carattere

Tipo di dato strutturato  
Il Record: Scheda con informazioni su un’oggetto , studente ecc.  
Struttura dato che mi permettere di raggruppare informazioni di tipo diverso.  
Es: nome Char  
 Età Int Campi del record  
 %Vittorie Reale  
Quindi il record avrà 1 campo del nome, 1 dell’età, 1 della % di vittorie .  
Quando voglio usare il record nei miei codici lo uso come entità unica, se voglio sapere quanti anni ha il giocatore, andrò a chiedere l’età del giocatore in posizione n, e anche per tutti i dati.  
Array di record esempio:  
Pos 1 : Roger Pos2: C  
 41 20  
 0,90 0,7  
Serve perché nel record posso mettere più valori diversi , non sono vincolato dall’array che permette di mettere solo lo stesso tipo di dati es: solo int, o reale o char soltanto.  
Possono esserci Record dentro altri Record , per esempio la data di nascita del giocatore che si trova dentro il record giocatore , anch’essa sarà un record ed è già presente in un record , perché contiene i numeri interi giorno , poi il carattere che li divide es: “/,.,-” mese , di nuovo il carattere, e l’anno.  
  
Record moneta:  
Peso Reale  
Valuta Array di carattere  
Data emissione Data  
Dimensione Reale  
Colori Array di caratteri  
  
**In c il record si chiama struct.**  
  
  
  
  
  
  
  
  
  
  
Function SommaVettore(Vettore1) = 1 INTERO DI TUTTI GLI ELEMENTI SOMMATI TRA LORO IN Vettore1.  
Function SommaVettori (Vettore1,Vettore2) output: VettoreS  
Record vettore  
Valore Array di interi  
Dimensione Intero >0



**Record Vettore formulazione base: (si parte sempre con la dimensione e il valore/carattere)**  
Valori Valori del vettore array di interi  
Dim Numero valori nel vettore intero>0  
**Senza vettore**  
Max = -infinito  
Pos= 1  
Mentre (Pos<=n)  
 Se(elemento di v in posizione Pos > Max)  
 Allora MAX= elemento di V in poosizione pos  
 Fine  
 Pos=Pos+1  
Fine  
**Con vettore**  
Max = - infinito  
Pos=1  
Mentre(Pos<= campo dim di v)  
 Se (elemento del campo valori di V in posizione pos > max)  
 Allora Max = elemento del campo valori di v in posizione pos  
 Fine  
 Pos=Pos+1  
Fine

MANIERE CORRETTE PER SCRIVERE VETTORI  
Elemento in posizione p del campo valori di v  
Campo valori di v  
campo dim di v  
 cancellare elemento del prof:  
input  
v vettore da cui cancellare 1 elemento, vettore  
valore, valore d1 v da cancellare, intero  
output  
v vettore con 1 elemento in meno vettore  
pseudo  
pos=1   
Mentre(pos <= campo dim di v) and (x=campo dim di v)  
Se(elemento in posizione pos del campo valori di v=valore)  
Allora   
 //Spostare a sinistra gli elementi del campo valori di V della posizione pos+1  
se(pos<campo dim di v)  
allora /sposta  
fine  
//decrementare campo dim (campo dim di V=campo dim di v-1)  
altrimenti pos=pos+1  
fine  
fine  
  
  
  
Algoritmo  
Descrivere l’algoriiavetmo (con pseudo)  
sintassi pseudocodice: parole chiave, operatori matematici, operatori logici, dati da descrivere (I/O), funzioni da usare.  
  
astrazione sui dati e astrazione funzionale  
permette di individuare per ogni dato della nostra soluzione quali sono le caratteristiche utili.

28/10/22  
Vettore Record  
Valori, valori presenti nel vettore, array di interi  
Dim, numero di valori del vettore, naturale>0  
**Trovare massimo del vettore**  
**Input**   
v , vettore di interi di cui cercare il massimo  
**Ouput**  
Max, valore massimo di v, intero  
**Pseudo**  
Max=- infinito  
Pos=1  
Mentre (Pos<= campoDim di V)  
 Se (max< elemento del campo valori di v in posizione pos)  
 Max= elemento del campo valori di v in posizione pos.  
 Fine  
 Pos=Pos+1  
Fine  
**Vettore array di interi**  
Max=- infinito   
Pos= 1  
Mentre (Pos<=N )  
 Se(Max<elemento di V in posizione pos)  
 Allora  
 Max = elemento di v in posizione pos  
 Fine  
 Pos=Pos+1  
Fine  
**Vettore come puntatore a interi  
Vettore come lista di interi**  
---  
Max=- infinito   
Pos= 1  
Mentre (Pos<=Dimensione vettore)  
 Se(Max<elemento corrente)  
 Allora  
 Max = elemento corrente  
 Fine  
 Pos=Pos+1  
Fine

**Leggere valore**  
Input  
v vettore  
p naturale >0  
output   
valore intero  
pseudo  
valore= elemento del campo valori di v in posizione p  
**Scrivere dimensione**  
Input  
v   
nuova dim  
output   
v  
pseudo  
campo dim di v= nuova dim  
--  
Max=- infinito   
Pos= 1  
Mentre (Pos<= Leggere Dimensione vettore(v))//con funzioni funziona  
 Se(Max<leggere valore (v,pos))  
 Allora  
 Max = Leggere valore (V,pos)  
 Fine  
 Pos=Pos+1  
Fine  
Leggere dimensione  
input  
v  
output  
d  
pseudo   
d= campo dim di v  
**Scrivere valore**  
**Input**  
V  
P  
Valore  
**Ouput**  
V  
**Pseudocodiece**  
elemento del campo valori di v in posizione p = valori

**Somma vettore**  
**input**  
v   
**output**  
somma  
**pseudo**  
somma=0  
pos=1  
mentre (pos < leggeredimensione(v))  
 somma=somma+leggerevalore(v,pos)  
 pos=pos+1  
fine  
SOMMA VETTORI  
input  
V1  
v2  
output  
vs  
pseudocodice  
pos=1   
mentre (pos<=leggere dimensione (v1))  
vs= scrivere valore (vs, leggere valore (v1,pos)+ leggere valore (v2,pos))  
pos=pos+1  
fine

09/11/22  
Dato un vettore a 2 dimensioni di interi, trovare il minimo tra i massimi delle righe.  
Input   
V, vettore a 2 dimensioni di interi, array a 2 di numeri di interi  
R, numero di righe di v naturale >0  
C, numero di colonne di V , naturale>0  
Output  
Minimo, minimo tra i valori massimi di ogni rigo. Intero  
**Pseudo**  
Minimo= + infinto  
pos=1   
mentre (pos<n) m= -infinito  
 m = trovaremassimo(v,m,pos) -> inc: pos2=1  
 se(m<minimo) mentre(pos2<n)  
 allora minimo= n Se(m< elemento di v in posizione pos e pos2)  
 fine allora  
 pos=pos+1 m=elemento di v in posizione pos e pos2;   
Fine fine  
 pos2=pos2+1  
  
libreria lmiits.h  
In C:  
#include <stdio.h>  
#include <stdlib.h>  
  
int main ()  
{   
int V[20] [10] oppure const int R=20; const int C=10 oppure #define R20; # define C10.  
int minimo;  
int m;  
int pos;  
int pos2;  
minimo = int\_max;  
pos=0;  
while(pos<c)  
{  
 if (m<V[Pos][Pos2]  
 {  
 m= V[Pos][Pos2];  
 }  
 Pos2=pos2+1;  
}  
if (mminimo)  
{minimmo=m;}  
pos=pos+1;  
}  
  
  
CON LE FUNZIONI:  
while(pos<R)  
{  
m=trovareMassimo (V,Pos,C);  
if(m<minimo)  
{  
minimo=m;  
}  
pos=pos+1;  
}  
return 0;  
}  
int trovareMassimo( int V[][],int v, int c)  
{   
int m;  
int pos;  
m= int\_min;  
pos=0;  
while (pos<c)  
{ IF(m <V[r][pos])  
 (m=v[r][pos];}  
pos=pos+1;  
}  
return m;  
}



RECORD IN Pseudo in c  
Pseudo:  
Orario Record  
-ore, int  
-giorni, int  
- tabella, Array di 2 dimensioni di lezioni  
C:   
struct Orario  
{ int ore;  
int giorni;  
lezioni Tabella [100] [100];  
}orario1;  
  
Pseudo:  
Vettore Record  
Valori, array di interi  
dim, intero  
C:  
Struct Vettore  
{  
int valori[100];  
int dim;  
}vettoreprova;  
  
Come si usa la struct in c:  
vettoreprova.valori[5]  
vettorediprova.dim;   
  
leggedimensione(vettoreprova);  
dim=vettoreprova.dim;  
return dim;  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
Struct modificareOrario (struct orario 0)  
{ int ore;  
 int giorno;  
 struct lezione l ;  
 printf (“msg”);  
scanf (“%d”,&ore);  
printf (“msg2”);  
scanf(“%d”,&giorno);  
l= Leggerelezionedamodificare(l);  
O= ScrivereLezione (O,ore,giorno,l);  
return 0;  
}  
  
Struct lezione Lezione da modificare (struct lezione l)  
{char aula;  
char materia,  
float ore-i;  
float ore-f;  
printf(“msg”);  
scanf(“%d”,&aula);  
l= ScrivereAula (l,Aula);  
Return l;  
}  
  
Scruvere Aula   
Input  
Lezione   
Aula  
Output  
lezione  
pseudo  
campo aula di lezione = aula  
IN C  
Struct lezione ScrivereAula (struct lezione l , char aula)  
{  
l.aula=aula;  
return l;  
}  
Per indirizzo :  
void lezione ScrivereAula (struct lezione \*l , char aula)  
{  
l->aula=aula;  
return;  
}

PUNTATORI  
Puntatori sono di vario tipo , intero, char, reale, record ma deve puntare a qualcosa altrimenti non serve a nulla.  
Puntatore ad intero intendo dire se io dichiaro :  
x puntatore intero (in memoria non ho nulla, nessuno spazio occupato, perché non ha ancora l’indirizzo di memoria)  
y intero (in memoria c’è uno spazio di memoria in grado di contenere 1 intero es: 2 byte allocato alla variabile y)  
  
int y;  
int \*x;:  
y= 10 ;  
x=&y;  
STRINGHE   
Int strlen (stringa) controlla la lunghezza stringa  
int strcpy (stringa1,stringa2) copia stringa 1 in 2  
int strcmp (stringa1,stringa2) confronta le 2 stringhe se sono uguali  
0 se sono ugali , se non è 0 non sono uguali.  
dichiarazione:  
char nome[10];  
nome[9]= “\0”; (aggiungere “\0”questo per indicare la stringa)  
ES:  
struct lezione  
{ char \*materia; //puntatori ad array  
 char \*aula; //  
 float ora\_inizio;  
 float ora\_fine;  
}  
scanf (“%s”, &stringa) il c inserisce da solo il simbolo di fine stringa “\0!  
scanf (“%c”,&stringa ) bisogna inserire manualmente il simbolo di fine stringa sennò andremo ad inserire un carattere.