

## RELAZIONE DELLA PROVA DI LABORATORIO DI INFORMATICA

Anno scolastico		Lab informatica	AULA n° 35
Data inizio svolgimento			
Progr. relazione	<input type="checkbox"/> primo trimestre		
	<input type="checkbox"/> secondo pentamestre		



### DATI DELLO STUDENTE

Cognome e Nome	
Classe e Sezione	

### PROBLEMA PROPOSTO

Problema n.ro		Nome del file relazione (1)	
		Nome del file progetto/sorgente (1)	
Titolo			
Breve descrizione			

### DATI DI INPUT DEL PROBLEMA PRINCIPALE (MAIN)

Nome variabile	Tipo dati (2)	Tipo Allocaz. (3)	Valori ammessi	Descrizione

### DATI DI OUTPUT DEL PROBLEMA PRINCIPALE (MAIN)

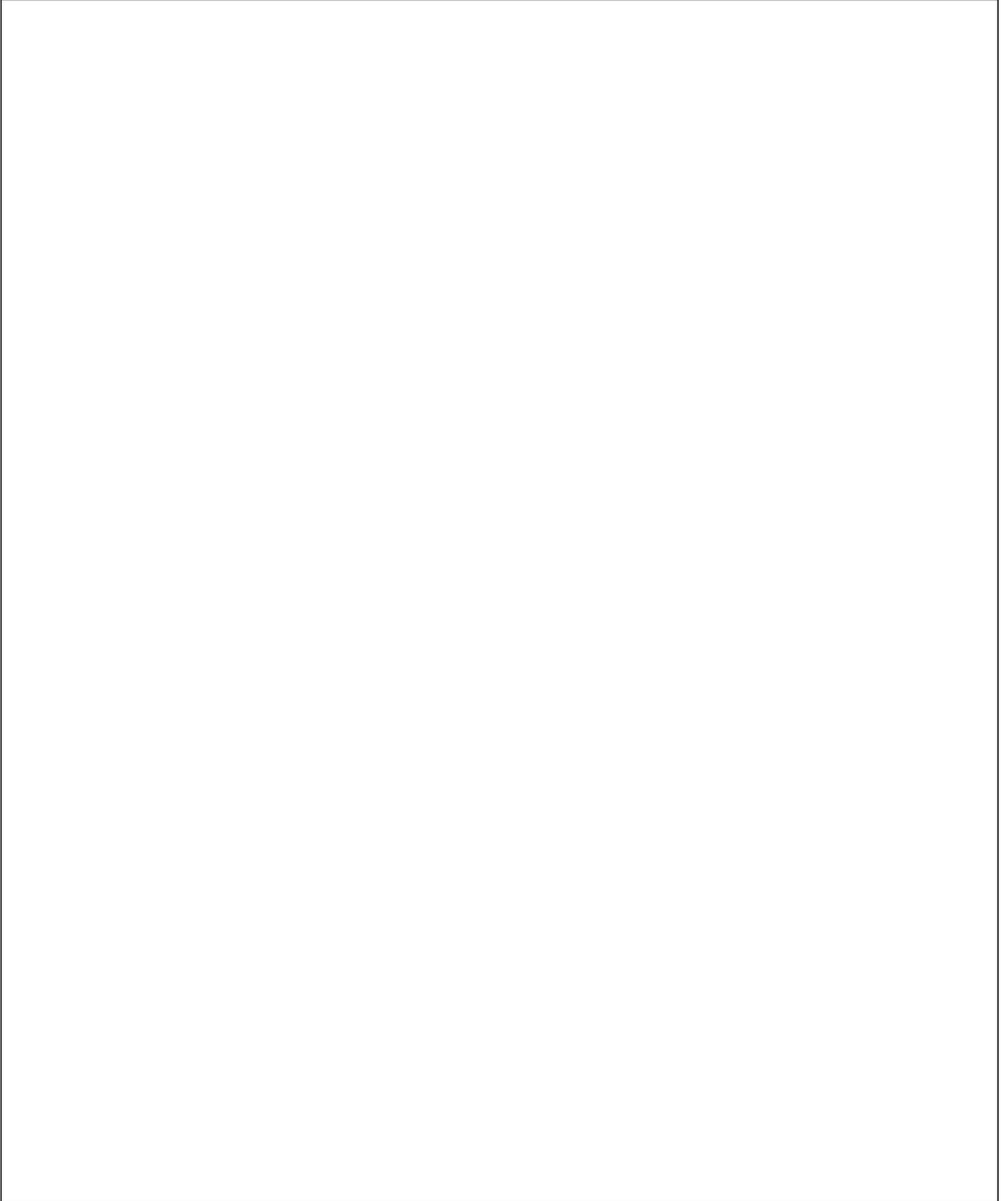
Nome variabile	Tipo dati	Tipo Allocaz.	Valori ammessi	Descrizione

### DATI DI ELABORAZIONE DEL PROBLEMA PRINCIPALE (MAIN)

Nome variabile oppure costante	Tipo dati	Tipo Allocaz.	Valori ammessi	Descrizione

**PSEUDOCODIFICA DELL'ALGORITMO RISOLUTIVO DEL PROBLEMA PRINCIPALE (MAIN)**

**DIAGRAMMA DI FLUSSO (FLOWCHART) DELL'ALGORITMO RISOLUTIVO DEL PROBLEMA (4)**



<b>SOTTOPROBLEMA (5)</b>				
<b>Nome del Sottoproblema</b>				
<b>Tipologia Sottoproblema</b>			Procedura <input type="checkbox"/>	Funzione <input type="checkbox"/>
<b>Tipo dati restituito esplicitamente nel nome</b> (La presente sezione va compilata solo in caso di Funzione)				
<b>Nome Variabile restituita</b>	<b>Tipo dati</b>	<b>Tipo Allocaz.</b>	<b>Valori ammessi</b>	<b>Descrizione</b>
<b>LISTA PARAMETRI FORMALI</b>				
<b>Nome parametro formale</b>	<b>Tipo dati</b>	<b>Tipo Passaggio (6)</b>	<b>Valori ammessi</b>	<b>Descrizione</b>
<b>DATI DI ELABORAZIONE DEL SOTTOPROBLEMA</b>				
<b>Nome variabile/costante</b>	<b>Tipo dati</b>	<b>Tipo Allocaz.</b>	<b>Valori ammessi</b>	<b>Descrizione</b>

**PSEUDOCODIFICA DELL'ALGORITMO RISOLUTIVO DEL SOTTOPROBLEMA**

**DIAGRAMMA DI FLUSSO (FLOWCHART) DELL'ALGORITMO RISOLUTIVO DEL SOTTOPROBLEMA**

**Data fine svolgimento**

**Firma dello studente**

## NOTE RELATIVE ALLA STESURA DEL DOCUMENTO

- (1) Ad esempio si potrebbe usare una nomenclatura che segua una convenzione del tipo  
<A.S.>\_<CLASSE>\_<COGNOMENOME>\_<PRG>.DOC  
che nel caso della relazione numero 1 dell'alunno MARIO ROSSI della 3H nell'anno scolastico 2005/2006 comporterebbe la creazione dei due file seguenti:
- file in formato WORD contenente la relazione:  
**0506\_3H\_ROSSIMARIO\_01.doc**
  - file contenente il progetto oppure il programma in linguaggio C:  
**0506\_3H\_ROSSIMARIO\_01.DEV** oppure **0506\_3H\_ROSSIMARIO\_01.C**
- (2) Nel caso di variabili strutturate di tipo **RECORD** oppure ARRAY [...] DI **RECORD** occorre dettagliare nel campo "Descrizione" anche il **nome** ed il **tipo dati** dei **CAMPI** costituenti il tipo RECORD
- (3) Da valorizzare con:
- **STAT** per indicare l'allocazione della variabile di tipo statico
  - **DIN** per indicare l'allocazione della variabile di tipo dinamico
- (4) Finchè non si sarà raggiunta una dimestichezza notevole con l'applicativo Microsoft Word il diagramma di flusso (o diagramma a blocchi) esplicativo del programma potrà essere redatto, dopo avere stampato la relazione compilata in ogni sua altra parte, a mano libera utilizzando una penna blu oppure nera ed un righello o squadretta.
- (5) Questa sezione va ripetuta tante volte per tutti i sottoproblemi nei quali verrà suddiviso il problema di partenza in accordo con la metodologia di progettazione di tipo top-down
- (6) Da valorizzare con:
- **VAL** per indicare che il passaggio del parametro dal problema al sottoproblema avviene per "valore" o BY VALUE
  - **RIF** per indicare che il passaggio del parametro dal problema al sottoproblema avviene per "riferimento" o BY REFERENCE

## STANDARD DI CODIFICA IN C DA UTILIZZARE

```

/*****
/*H  ANNO SCOLASTICO:          <anno scolastico>          */
/*H                                     */
/*H  AUTORE:                   <nome e cognome autore>    */
/*H                                     */
/*H  CLASSE:                   <classe frequentata>       */
/*H                                     */
/*H  DATA CREAZIONE:         */
/*H                                     */
/*H  DATA ULTIMA MODIFICA:   */
/*H                                     */
/*****/

/*****
/*  Sezione inclusione librerie di sistema                */
/*****/
#include <stdio.h>
#include <stdlib.h>

/*****
/*  Sezione di inclusione eventuali dichiarazioni di funzioni, tipi di dato          */
/*  variabili "utente" inseriti in altri FILE                                         */
/*****/
#include "t000typ.h

/*****
/*  Sezione di dichiarazione eventuali tipi GLOBALI "utente" utilizzati             */
/*****/
typedef int  miointero;

/*****
/*  Sezione di dichiarazione delle costanti "utente" utilizzate                     */
/*****/
#define  TRUE  1

/*****
/*  Sezione di dichiarazione delle risorse interne o esterne                         */
/*  (funzioni e variabili globali) */
/*****/
[extern] int  miafunzione([lista_parametri]); ("extern" si usa se la funzione è codificata in un altro file)
[extern] int  sconto;                       ("extern" si usa se la variabile è stata dichiarata in un altro file)

```

### NOTA BENE

Ad eccezione dell'istruzione main l'indentazione (ossia il numero di caratteri vuoti o blank da digitare prima) del simbolo "{ " oppure "}" deve essere pari a 3.

```
/* **** */
/* Programma principale */
/* **** */
int main (int argc, char *argv[])
{
  <sezione dichiarativa>
  <corpo del programma>
  return 0;
}

/* **** */
/* Sottoprogramma (corpo della funzione) */
/* (da ripetere tante volte per quante sono le funzioni individuate) */
/* **** */
<istruzione di definizione funzione con lista parametri formali>
{
  <sezione dichiarativa>
  <corpo della funzione>
  return <nome_variabile_restituita>;
}
```

### Istruzione WHILE-DO

```
/* **** */
/* <Commento della logica del ciclo iterativo WHILE> */
/* **** */
while (<condizione>)
{
  /* Blocco Istruzioni */
}
```

### Istruzione DO-WHILE

```
/* **** */
/* <Commento della logica del ciclo iterativo DO-WHILE> */
/* **** */
do
{
  /* Blocco Istruzioni */
}
while (<condizione>;);
```

### Istruzione FOR

```
/* **** */
/* <Commento della della logica del ciclo iterativo FOR> */
/* **** */
for (<inizializzazione indice>; <condizione di ciclo>; <incremento/decremento dell'indice>)
{
  /* Blocco Istruzioni */
}
```

### Istruzione IF

```
/* **** */
/* <Commento della logica di percorrenza ramo VERO> */
/* **** */
if (<condizione>)
{
  /* Blocco Istruzioni ramo VERO */
}
```

### Istruzione IF-ELSE

```
/* **** */
/* <Commento della logica di percorrenza ramo VERO> */
/* **** */
if (<condizione>)
{
  /* Blocco Istruzioni ramo VERO */
}
/* **** */
/* <commento della logica di percorrenza ramo FALSO> */
/* **** */
else
{
  /* Blocco Istruzioni ramo FALSO */
}
```

### Istruzione IF-ELSEIF

```
/* **** */
/* <Commento della logica di percorrenza dell'istruzione complessiva>
   */
/* **** */
if (<condizione1>)
{
  /* Blocco Istruzioni se condizione1 è VERA */
}
/* **** */
/* <eventuale commento > */
/* **** */
elseif (<condizione2>)
{
  /* Blocco Istruzioni se condizione2 è VERA (ma condizione1 è FALSA) */
}
/* **** */
/* <eventuale commento > */
/* **** */
elseif (<condizione3>)
{
  /* Blocco Istruzioni se condizione3 è VERA (ma condizione1 è FALSA e condizione 2 è FALSA) */
}
.....
/* **** */
/* <eventuale commento > */
/* **** */
else
{
  /* Blocco Istruzioni se tutte le condizioni impostate risultano false */
}
```

### Istruzione SWITCH

```
/* **** */
/*      <Commento della logica complessiva dello switch>      */
/* **** */
switch (<nome_variabile>)
{
/* **** */
/*      <Commento rispetto a VALORE-1 >      */
/* **** */
case <VALORE-1>:
{
/* Blocco Istruzioni 1 */
break;
}
/* **** */
/*      <Commento rispetto a VALORE-2 >      */
/* **** */
case <VALORE-2>:
{
/* Blocco Istruzioni 2 */
break;
}
.....

/* **** */
/*      <Commento rispetto a VALORE-N >      */
/* **** */
case <VALORE-N>:
{
/* Blocco Istruzioni N */
break;
}
/* **** */
/*      <Commento rispetto alla condizione di altrimenti >      */
/* **** */
default:
{
/* Blocco Istruzioni DEFAULT*/
break;
}
}
```