

Esame di Laboratorio di Informatica: Programmazione I

11 febbraio 2003

Prof. Sebastiani

Codice:

Nome	Cognome	matricola

La directory 'esame' contiene 4 sotto-directory: 'uno', 'due', 'tre' e 'quattro'. Le soluzioni vanno scritte negli spazi e nei modi indicati esercizio per esercizio.

NOTA: il codice dato non può essere modificato.

Modalità d'Esame

Durante la prova gli studenti sono vincolati a seguire le regole seguenti:

- Non è consentito l'uso di alcun libro di testo o fotocopia. In caso lo studente necessitasse di carta (?), gli/le verranno forniti fogli di carta bianca su richiesta, che dovranno essere riconsegnati a fine prova. È consentito l'uso di una penna. Non è consentito l'uso di alcuno strumento calcolatore.
- È vietato lo scambio di qualsiasi informazione, orale o scritta. È vietato guardare nel terminale del vicino.
- È vietato l'uso di telefoni cellulari o di qualsiasi strumento elettronico.
- È vietato allontanarsi dall'aula durante la prova, anche se si ha già consegnato. (Ogni necessità fisiologica va espletata PRIMA dell'inizio della prova.)
- È vietato qualunque accesso, in lettura o scrittura, a file esterni alla directory di lavoro assegnata a ciascun studente. Le uniche operazioni consentite sono l'apertura, l'editing, la copia, la rimozione e la compilazione di file all'interno della propria directory di lavoro.
- Sono ovviamente vietati l'uso di email, ftp, ssh, telnet ed ogni strumento che consenta di accedere a file esterni alla directory di lavoro. Le operazioni di copia, rimozione e spostamento di file devono essere circoscritte alla directory di lavoro.
- Ogni altra attività non espressamente citata qui sopra o autorizzata dal docente è vietata.

Ogni violazione delle regole di cui sopra comporterà automaticamente l'annullamento della prova e il divieto di accesso ad un certo numero di appelli successivi, a seconda della gravità e della recidività della violazione.

NOTA IMPORTANTE: DURANTE LA PROVA PER OGNI STUDENTE VERRÀ ATTIVATO UN TRACCIATORE SOFTWARE CHE REGISTRERÀ TUTTE LE OPERAZIONI ESEGUITE (ANCHE ALL'INTERNO DELL'EDITOR!!). L'ANNULLAMENTO DELLA PROVA DI UNO STUDENTE POTRÀ AVVENIRE ANCHE IN UN SECONDO MOMENTO, SE L'ANALISI DELLE TRACCE SOFTWARE RIVELASSERO IRREGOLARITÀ.

1 Dato il seguente file:

```
//file main.cc
#include<iostream>

void cambio (char *, char *, char *);

void main(){
    char a,b,c;
    char *w =&a;
    char *x =&b;
    char *y =&c;

    cout << "inserisci il primo valore \n";
    cin >> *x;
    cout << "inserisci il secondo valore \n";
    cin >> *y;
    cout << "inserisci il terzo valore \n";
    cin >> *w;
    cout << "i valori inseriti: " << *x << " " << *y << " " << *w << endl;
    cambio(x,y,w);
    cout << "i valori scambiati: " << *x << " " << *y << " " << *w << endl;

}

//scrivere il proprio codice al di sotto di questa linea
//write your code under this line
```

scrivere il corpo della funzione **cambio** in modo tale che realizzi lo scambio dei *valori puntati* dai tre parametri nel modo illustrato dal seguente esempio:

valori inseriti: a='m', b='n', c='o',

valori scambiati dopo la chiamata della funzione cambio: a='o', b='m', c='n'.

English version

Given the above file, write the body of function **cambio** in such a way to implement the shift of the *values pointed* by the three parameters as figured by the following example:

inserted values: a='m', b='n', c='o',

shifted values after function call: a='o', b='m', c='n'.

2 Dato il seguente file:

```
//file main.cc
#include<iostream.h>

int* quoziente(int* v1,int* v2,int dim);

void leggivettore(int* v,int dim){
    for(int i=0;i<dim;i++)
        cin >> v[i];
}

void stampavettore(int* v,int dim){
    for(int i=0;i<dim;i++)
        cout << v[i] << " ";
    cout << endl;
}

void main(){
    int v1[7];
    int v2[7];
    int *ris;
    cout << "V1 ";
    leggivettore(v1,7);
    cout << "V2 ";
    leggivettore(v2,7);
    ris=quoziente(v1,v2,7);
    cout << "V1 ";
    stampavettore(v1,7);
    cout << "V2 ";
    stampavettore(v2,7);
    cout << "ris ";
    stampavettore(ris,7);
}

//Scrivete il vostro codice al di sotto di questa linea
//Write your code under this line
```

implementare la funzione **quoziente** in modo tale che prenda due vettori **v1** e **v2** della stessa dimensione, la loro dimensione **dim** e restituisca un nuovo vettore il cui *i*-esimo elemento sia la quoziente degli *i*-esimi elementi di **v1** e **v2**.

NOTA: non usare variabili globali o statiche.

English version

Given the above file **main.cc**, implement the function **quoziente** in such a way to take two arrays **v1** and **v2** of the same size, their size **dim** and to return a new array whose *i*-th element must be the quotient of the *i*-th elements of **v1** and **v2**.

NOTE: don't use global or static variables.

3 Dato il seguente file:

```
//file main.cc
#include<iostream.h>
#include<string.h>

struct persona{
    char* nome;
    char* cognome;
    char* email;
    char* professione;
};

int ric_bin(persona v[],char* el,int i,int f);

void stampa_persona(persona &p){
    cout << "======" << endl;
    cout << p.nome << endl;
    cout << p.cognome << endl;
    cout << p. email << endl;
    cout << p.professione << endl;
    cout << "======" << endl;
}

void main(){
    persona e0={"Ascanio","Sforza","asforza@science.unitn.it","cardinale"};
    persona e1={"Girolamo","Savonarola","gsavo@science.unitn.it","frate"};
    persona e2={"Cristoforo","Colombo","ccolombo@science.unitn.it","navigatore"};
    persona e3={"Rodrigo","Borgia","alessandroVI@science.unitn.it","papa"};
    persona e4={"Raffaello","Sanzio","rsanzio@science.unitn.it","pittore"};

    persona V[5]={e0,e1,e2,e3,e4};

    int ret=ric_bin(V,"papa",0,5);
    if(ret!=-1)
        cout << "Non trovato" << endl;
    else
        stampa_persona(V[ret]);
}

//Scrivete il vostro codice al di sotto di questa linea
//Write your code under this line
```

scrivere l'implementazione della funzione **ric_bin** in modo tale che, usando una ricerca binaria/dicotomica e sapendo che **v** è ordinato alfabeticamente rispetto a **professione**, restituisca l'indice dell'elemento dell'array con il campo **professione** uguale a **el**, -1 altrimenti.

English version

Given the above file, write the implementation of the function **ric_bin** in such a way that, by using a binary/dichotomizing search and knowing that **v** is sorted alphabetically w.r.t. **professione**, it returns the index of the array's element with field **professione** equal to **el**, -1 otherwise.

4 Dati i seguenti file:

```
//file mazzo.h
#ifndef MAZZO_H
#define MAZZO_H

#include<iostream.h>

enum retval {FAIL,OK};
enum seme {BASTONI, DENARI, COPPE, SPADE};
enum valore {uno,due,tre,quattro,cinque,sei,sette,fante,cavallo,re};
struct carta {seme s; valore n;};

void stampa_la_carta(carta);
void dai_carta(carta &);

void init();
retval metti_nel_mazzo(const carta);
retval toglid_dal_mazzo(carta &);

#endif

//file mazzo.cc
#include"mazzo.h"

void stampa_la_carta(carta c)
{cout << (int)c.n+1 << " di ";
  cout << (((int)c.s==0? "BASTONI":(((int)c.s==1)? "DENARI":
    ((int)c.s==2? "COPPE": "SPADE")))<< endl;
};

void dai_carta(carta & c)
{
  int num1 = -1; int num2 = 5;

  while ( (num1 < 1) || (num1 > 10))
  {cout << "inserisci un numero intero(come carta napoletana da 1 a 10) \n";
    cin >> num1;
    if ((num1 < 1) ||(num1 > 10)) cout << "hai sbagliato ad inserire il numero \n";
  };
  while ( (num2 < 1) || (num2 > 4))
  {cout <<
    "inserisci un numero(come seme:BASTONI=1,DENARI=2,COPPE=3,SPADE=4) \n";
    cin >> num2;
    if ((num2 < 1) ||(num2 > 4)) cout << "hai sbagliato ad inserire il seme \n";
  };
  c.n = (valore)(num1-1);
  c.s = (seme)(num2-1);
};

/*Scrivete il vostro codice al di sotto di questa riga*/
/*Write your code under this line*/
```

```

//file main.cc
#include"mazzo.h"

int main(){
    carta mia;
    char res;
    init();
    do {
        cout << "\nOperazioni possibili:\n"
             << " mettere una carta nel mazzo (m)\n"
             << " prendere una carta dal mazzo (p)\n"
             << " Fine (f)\n";
        cin >> res;
        switch (res) {
            case 'm':
                dai_carta(mia);
                metti_nel_mazzo(mia);
                break;
            case 'p':
                if (togli_dal_mazzo(mia)==FAIL)
                    cout << "mazzo vuoto!\n";
                else stampa_la_carta(mia);
                break;
            case 'f':
                break;
            default:
                cout << "Opzione errata\n";
        }
    }while (res != 'f');
}

```

Implementare un mazzo di carte da briscola con una coda di carte e scrivere il corpo delle funzioni **metti_nel_mazzo**, **togli_dal_mazzo**, **init** in modo tale che abbiano le seguenti funzionalità:

- **init** inizializza una coda vuota.
- **metti_nel_mazzo** inserisce un elemento di tipo carta nella coda. Restituisce OK se l'inserimento è riuscito, FAIL altrimenti.
- **togli_dal_mazzo** preleva, cancellandolo, il primo elemento inserito nella coda (politica FIFO). Restituisce FAIL se il mazzo è vuoto, OK altrimenti.

La coda può essere implementata tramite array o puntatori, a scelta (nel primo caso usare un array di 40 carte).

Given the above files, write the implementation of a pack of napoletane cards by mean of a queue of cards. Then, give the implementation of the above functions with the following features:

- **init** initializes an empty queue.
- **metti_nel_mazzo** puts an element of type carta in the queue. It returns OK if the operation is succesful, FAIL otherwise.
- **togli_dal_mazzo** takes an the first element (FIFO policy) of type carta inserted into the queue and deletes it. It returns FAIL if the card pack is empty, OK otherwise.

The queue can be implemented using arrays or pointers, student's choice (in the first case use an array of 40 cards).

- 5 Gli antichi romani rappresentavano i numeri con il seguente metodo.¹ Partendo dalle migliaia fino alle unità, rappresentavano il numero di migliaia, cinque-centinaia, centinaia, cinquantine, decine, cinque ed unità con una sequenza di un numero corrispondente di caratteri M, D, C, L, X, V, I rispettivamente:

Cifre romane:	M	D	C	L	X	V	I
Valore corrispondente:	1000	500	100	50	10	5	1

Esempi:

Numero	7	58	1666	2002	4972
Numero romano corrisp.:	VII	LVIII	MDCLXVI	MMII	MMMMDCCCCLXXII

Dato il seguente file:

```
#include <iostream.h>

// numero di caratteri sufficiente a memorizzare ogni
// numero romano in [1...9999]
const int maxdim = 100;

// converte un numero intero in un numero romano
char * converti(int numero);

void main()
{
    int numero;
    char * numeroromano;

    do {
        cout << "Dammi il numero da convertire [1...9999]: ";
        cin >> numero;
        numeroromano = converti(numero);
        cout << "Il numero romano corrispondente a " << numero << " e' "
             << numeroromano << endl;
    } while (numero >= 1 && numero <= 9999);
}

// APPENDERE QUI SOTTO LA DEFINIZIONE DELLA FUNZIONE converti
```

scrivere la definizione della funzione **converti** che prende in ingresso un numero compreso tra 1 e 9999 e restituisce una stringa di caratteri che rappresenti il numero romano corrispondente.

English version

The ancient Romans used to represent integer numbers in the following way. Starting from thousands down to units, they represented the number of thousands, five-hundreds, hundreds, fifties, tens, fives and units by a corresponding sequence of characters M, D, C, L, X, V, I respectively. (See examples above). Given the main file above, write the definition of the function **converti** which takes as input an integer number and returns a string of characters representing the corresponding roman number. ²

¹Per i puristi del latino: per semplicità non si consideri in questo caso l'antica usanza di scrivere IV per 4, IX per 9, XL per 40, XC per 90, CD per 400, CM per 900.

²For Latin purists: for simplicity you we do not consider here the ancient habit of writing IV for 4, IX for 9, XL for 40, XC for 90, CD for 400, CM for 900.