

1. Esercizi risolti

Esempio 1

Scrivere un programma di simulazione delle nascite, per il calcolo della frequenza dei maschi e delle femmine. L'ampiezza del campione è immessa da input e la sequenza dei dati deve essere memorizzata in un array per usi successivi.

Il campione viene generato in modo casuale mediante la funzione **rand()** che restituisce un valore tra 0 e 99999. Attraverso il modulo 10 (%) si ottiene un valore tra 0 e 9 e si può ipotizzare che un valore maggiore di 5 indichi maschio ('m'), altrimenti indichi femmina ('f').

L'ampiezza *amp* letta da input, consente di allocare dinamicamente l'array *Sesso* con l'istruzione

```
char * Sesso = new char [amp];
```

L'analisi del campione è svolta dalla funzione:

```
void analisi (char sex[], int ampiezza);
```

mentre la costruzione dei contatori dei maschi e delle femmine, *cm* e *cf*, viene realizzata passandoli per indirizzo alla funzione:

```
void conteggio (char sex[], int len, int &cm, int &cf);
```

Al termine del programma occorre deallocare l'array *Sesso* mediante la chiamata **delete [] Sesso**. Una possibile soluzione è descritta dal programma di *Listato 2.3*.

```
/* Statistiche.cpp */
#include<iostream>
using namespace std;
void conteggio (char sex[], int len, int &cm, int &cf);
void analisi (char sex[], int ampiezza);
int main()
{ int amp, /* ampiezza del campione */
  cf=0, /* numero massimo di femmine */
  cm=0; /* contatore di maschi */
  char * Sesso; /* per immettere il campione */
  cout<<"Generazione che usa il timer come seme"<<endl;
  srand(time(0)); /* generazione automatica */
  cout<<"Immetti l'ampiezza del campione: ";
  cin>>amp;
  Sesso=new char[amp];
  analisi(Sesso, amp);
  conteggio(Sesso, amp, cm, cf);
  cout<<"Numero di maschi..... = "<<cm<<endl;
  cout<<"Numero di femmine..... = "<<cf<<endl;
  delete [] Sesso;
  system("pause");
  return 0;
}
void conteggio(char sex[], int len, int &cm, int &cf)
{ /* costruisce le variabili cm e cf ... */
  int i; /* ...passate per indirizzo */
  for(i=0;i<len;i++)
    if (sex[i]=='m')
      cm++;
    else
      cf++;
}
```

```

void analisi (char sex[], int ampiezza)
{
  int i;
  for (i=1; i<=ampiezza; i++)
  {
    if (rand()%10>=5)          /* rappresenta 'maschio' */
      sex[i]='m';
    else sex[i]='f';          /* rappresenta 'femmina' */
  }
}

```

Esempio 2

Scrivere un programma che allochi dinamicamente un array di interi e che consenta di eseguire su esso le operazioni seguenti:

- sostituire l'elemento in posizione p con un nuovo valore v;
- eliminare con shift a sinistra, l'elemento in posizione p

I valori di v e di p vengono immessi da input.

Il programma si basa su un menu in cui è possibile scegliere diverse opzioni:

1. creazione dell'array;
2. modifica di una posizione;
3. eliminazione di una posizione;
4. fine programma.

L'array viene allocato dinamicamente in base alla dimensione richiesta e gestito mediante indice. Si prevede una funzione di stampa che, man mano che vengono eseguite le varie operazioni, stampa la sequenza di valori presenti. L'utente deve vedere le posizioni logiche partire da 1, quindi è necessario prevedere un'opportuna gestione degli indici, che tenga anche conto di eventuali accessi a posizioni inesistenti dell'array. Al termine, occorre prevedere la deallocazione della struttura dati. In *Listato 2.4* è presentata una possibile soluzione al problema che fa uso di funzioni.

OperazioniArrayDinamico.cpp

```

#include<iostream>
#include<conio.h>
using namespace std;

void modifica(int *vet, int ampiezza, int p, int v);
void carica (int *vet, int ampiezza);
void elimina (int *v, int p, int d);
void stampa (int vet[], int ampiezza);
int main()
{
  int   dim,          /* ampiezza del vettore */
        p,           /* posizione da elaborare */
        v,           /* valore da sostituire */
        *valori,
        scelta;

  cout<<"Generazione che usa il timer come seme"<<endl;
  srand(time(0));    /* generazione automatica */
  do
  {
    cout<<"*** Operazioni ***"<<endl;
    cout<<"Creazione vettore (1)"<<endl;
    cout<<"Modifica posizione(2)"<<endl;
    cout<<"Elimina posizione (3)"<<endl;
    cout<<"Fine programma (0)"<<endl;
    cout<<"Scelta: ";
    cin>>scelta;
    switch (scelta)

```

```

        {
            case 1: cout<<"Immetti l'ampiezza del vettore: ";
                    cin>>dim;
                    valori= new int[dim];
                    carica(valori, dim);
                    break;
            case 2: cout<<"Posizione da modificare (1-<<dim<<)"<<endl;
                    cin>>p;
                    if (p>dim || p<1)
                        cout<<"Posizione inesistente"<<endl;
                    else
                    {
                        cout<<"Immetti il nuovo valore: ";
                        cin>>v;
                        modifica(valori, dim, p, v);
                    }
                    stampa(valori, dim);
                    break;
            case 3: cout<<"Posizione da eliminare(1-<<dim<<)"<<endl;
                    cin>>p;
                    if (p>dim || p<1)
                        cout<<"Posizione inesistente"<<endl;
                    else
                    {
                        elimina(valori, p, dim);
                        dim--;
                    }
                    stampa(valori, dim);
                    break;
        }
    } while(scelta!=0);
if (!valori) delete []valori;
return 0;
}
void modifica(int *vet, int ampiezza, int p, int v)
{
    vet[p-1]=v;
}
void carica(int *vet, int ampiezza)
{
    int i;
    for (i=0; i<ampiezza; i++)
    {
        cout<<"Valore: ";
        cin>>vet[i];
    }
}
void elimina (int *v, int p, int d)
{
    int i;
    for(i=p-1; i<d-1; i++)
        v[i]=v[i+1];
}
void stampa (int *vet, int ampiezza)
{
    int i;
    for (i=0; i<ampiezza; i++)
        cout<<"Valore: "<<vet[i]<<endl;
}
}

```

Esempio 3

Si deve ripetere a ciclo continuo il calcolo della media aritmetica di misure di tipo reale relative a vari campioni di ampiezza differente. Scrivere un programma che realizzi quanto richiesto.

La soluzione può essere ottenuta allocando dinamicamente un array *s* in modo che, per ogni campione, venga dimensionato ad un valore uguale all'ampiezza di questo. Il ciclo termina quando l'utente, alla richiesta di ripetizione, digita il carattere 'N'.

Nel listato seguente è mostrata una possibile soluzione al problema.

MediaMisure.cpp

```

#include<iostream>
using namespace std;
int main()
{
    float *s=NULL, media;
    int nmisure, i;
    char fine='S';
    while (toupper(fine)!='N')
    {
        media=0;
        cout<<"Numero di misurazioni: ";
        cin>>nmisure;
        if (s) /* se spazio già allocato... */
            delete []s; /* ...dealloca... */
        s=new float[nmisure]; /* ...alloca */
        if (s==NULL) /* non c'è abbastanza memoria */
            exit(1);
        cout<<"La prova prevede "<<nmisure<<" misure"<<endl;
        for(i=0; i<nmisure; i++)
        {
            cout<<"Valore: ";
            cin>>s[i];
            media+=s[i];
        }
        cout<<"La media e' "<<(float)(media/nmisure)<<endl;
        cout<<"Continuare [S/N] ?";
        cin>>fine;
    }
    delete []s;
    return 0;
}

```

2. Esercizi applicativi

1. Determinare l'output dei seguenti frammenti di codice

```
a. #include <iostream>
#include <cstdlib>
using namespace std;
int main ()
{
    int t, *p;
    p=new int [40];
    if (!p)
        cout <<"Memoria esaurita!";
    else
        for (t=0; t<40; t++)
            p[t]=t;
        for (t=0; t<40;t++)
            cout<<p[t];
        ....
    delete [] p;
}
```

```
2. #include <iostream>
using namespace std;
#define RIGHE 5
#define COLONNE 2
int main()
{ int i,j,c=0,*matrice;
  matrice=new int[RIGHE*COLONNE];
  for(i=0;i<RIGHE;i++)
    for (j=0;j<COLONNE;j++)
      { matrice[i*COLONNE+j]=c;
        c++;
      }
  for (i=0;i<RIGHE;i++)
  {
    for(j=0;j<COLONNE;j++)
      cout<<matrice[i*COLONNE+j]<<" ";
    cout<<endl;
  }
  delete [] matrice;
  system ("Pause");
  return 0;
}
```

3. Individuare gli errori sintattici e/o logici nei seguenti frammenti di codice e, dopo averli corretti, descriverne l'effetto.

```
a. #include <iostream>
#include <cstdlib>
using namespace std;
int main()
{...
  char *s, int n;
  cinn>>n;
  s=new int[n];
  gets(s);
  ...
  delete [ ] s;
```

- b. **#include<iostream>**
#include <cstdlib>
using namespace std;
int main()
{
 int i, n=3;
 string parola;
 char *v[n];
 for (i=0; i<n; i++)
 {
 v[i]= **char**[10];
 cin>>v[i];
 cout<<v[i];
 }
 system ("pause");
 return 0;
}
- c. **#include <iostream>**
#include <cstdlib>
using namespace std;
int main()
{ ...
 int p, t;
 p= new int[40];
 for (t=0; t<n; t++)
 { ...
 p[t]=t;
 }
 delete p;

}