ARRAY E STRINGHE

ARRAY

ESERCIZIO: SHIFT ARRAY (1)

- Utilizzando le funzioni scrivere un programma che:
  - *genera una sequenza di N = 20 numeri interi e li memorizza in un array* (… riempi_array(…))
  - *visualizza il contenuto dell'array* (… stampa_array(…))
  - *esegue uno spostamento (shift) a sinistra di una posizione del contenuto dell'array* (… sn_shift_array(…))
  - Esempio:
    - vettore = 1 10 15 18
    - il programma deve generare il vettore: 10 15 18 0
  - *visualizza il contenuto dell'array* (… stampa_array(…))

ESERCIZIO: SHIFT ARRAY (2)

```
#include <stdio.h>
#include <stdlib.h>

void riempi_array(int * vet,  const int N);
void stampa_array(int * vet,  const int N);
void sn_shift_array(int * vet,  const int N);

int main(void){
  const int N = 20 ; // dimensione massima del vettore
  int vet[N] ; //  sequenza di numeri interi

  riempi_array(vet, N);
  stampa_array(vet, N);
  sn_shift_array(vet, N);
  stampa_array(vet, N);
}
```

ESERCIZIO: SHIFT ARRAY (3)

```
void riempi_array(int * vet,  const int
  N){
  printf("Inserisci una sequenza di %d
  numeri\n", N) ;
  int i;

  for ( i=0; i<N; i++ ){
     printf("Elemento %d: ", i+1) ;
     scanf("%d", &vet[i]) ;
  }
  printf("\n")
}
```

ESERCIZIO: SHIFT ARRAY (4)

```
void stampa_array(int * vet, int N){
  printf("La sequenza inserita e' la
  seguente\n") ;

  int i;
  for ( i=0; i<N; i++ )
    printf("Elemento %d: %d\n", i+1,
  vet[i]) ;
  printf("\n") ;
}
```

ESERCIZIO: SHIFT ARRAY (5)

```
void sn_shift_array(int * vet, int N){
  int i;
  // copia nella cella vet[i] il contenuto
  della cella vet[i+1]
  for ( i=0; i<N-1; i++ ){
     vet[i] = vet[i+1] ;
}

// assegna il valore alla cella vet[N-1]
vet[N-1] = 0 ;
}
```

STRINGHE

STRINGHE (1)

- Scrivere un programma in linguaggio C che
  - legga una frase introdotta da tastiera. La frase è terminata all'introduzione del carattere di invio. La frase contiene sia caratteri maiuscoli che caratteri minuscoli, e complessivamente al massimo 100 caratteri.
  - stampi a schermo la frase letta.
  - stampi a video per ognuna delle lettere dell'alfabeto, il numero di volte che la lettera compare nella stringa

STRINGHE (1)

```
void stampa(char* s);
void calcolo_lettere(char * s, int *
  alfabeto, int N);
void stampa_array(int * alfabeto, int N);

main () {
    // dichiarazione variabili
    char stringa[100];
    int alfabeto[26];

    // lettura stringhe da tastiera
    printf("inserisci una stringa: ");
    scanf("%s", stringa);
    stampa(stringa);  // già vista
```

STRINGHE (2)

```
  // inizializzo array
  for(i = 0; i < N; i++)
    alfabeto[i] = 0;

  calcolo_lettere(stringa, alfabeto, N);
  stampa_array(alfabeto, N);
}
```

STRINGHE (3)

```
void stampa(char* s){
  int i=0;
  while (s[i]!='\0'){// fino a quando non
  c'e' il carattere \n
     printf("%c", s[i]);
     i++;
  }
  printf("\n");
}
```

STRINGHE (4)

```
void calcolo_lettere(char * s, int * contatori, int
  N){
  i = 0, posizione_alfabeto=0;
  while (s[i]!='\0'){
     if (s[i] >= 'A' && s[i] <= 'Z' ){
        posizione_alfabeto = s[i] - 'A' ;
        contatori[posizione_alfabeto] ++ ;
     }
     else if(s[i] >= 'a' && s[i] <= 'z'){
        posizione_alfabeto = s[i] - 'a' ;
        contatori[posizione_alfabeto] ++ ;
     }
  }
}
```

ESERCIZIO (1)

- Variante del precedente programma: deve stampare a video anche
  - il numero di consonanti presenti nella stringa
  - il numero di vocali presenti nella stringa.

```
main(){
   ...
   printf("n consonanti = %d ",
   n_consonanti(alfabeto, N));
   printf("nvocali = %d ", n_vocali(alfabeto, N));
   ...
}
```

ESERCIZIO (2)

```
int n_consonanti(char * stringa){
  int consonanti = 0;
  while (s[i]!='\0'){
     if(s[i] >= 'a' && s[i] <= 'z')
        if(s[i] - 'a' != 0 && s[i] - 'e' != 0 && s[i]
   - 'i' != 0
        && s[i] - 'o' != 0 && s[i] - 'u' != 0)
           consonanti++
     else if(s[i] >= 'A' && s[i] <= 'Z')
        if(s[i] - 'A' != 0 && s[i] - 'E' && s[i]  -
  'I' != 0
        && s[i] - 'O' != 0 && s[i] - 'U' != 0)
           consonanti++
  }
  return consonanti;
}
```

ESERCIZIO (3)

```
int n_vocali(char * stringa){
  int vocali = 0;
  while (s[i]!='\0'){
     if(s[i] >= 'a' && s[i] <= 'z')
        if(s[i] - 'a' == 0 || s[i] - 'e' == 0 || s[i]
   - 'i' == 0
        || s[i] - 'o' == 0 || s[i] - 'u' == 0)
           vocali++
     else if(s[i] >= 'A' && s[i] <= 'Z')
        if(s[i] - 'A' == 0 || s[i] - 'E' || s[i]  -
  'I' == 0
        || s[i] - 'O' == 0 || s[i] - 'U' == 0)
           vocali++
  }
  return vocali;
}
```

ESERCIZIO (4)

Scrivere una funzione ricorsiva che riceva in input due stringhe es Nome e Cognome e ritorni 1 se le due stringhe sono uguali, 0 altrimenti.

int uguale (char nome[], char cognome[])

ESERCIZIO (4)

```
#include <stdio.h>
int uguale (char nome[], char cognome[])
{
  int i=0;
  if(nome[i]!=cognome[i]) return 0;
  if(nome[i]=='\0' && cognome[i]=='\0') return 1;
  return uguale(nome+1,cognome+1);
}
int main(){

char nome[20];
char cognome[20];
printf("Qual e' il tuo nome?\n");
scanf("%s",nome);
printf("Qual e' il tuo cognome?\n");
scanf("%s",cognome);

 int test=uguale(nome,cognome);
 printf ("\nrisultato del confronto: %d", test);

if(test==0) printf("\nNOME E COGNOME DIVERSI: CIAO %s %s !",nome,cognome);
    else printf("\nNOME E COGNOME UGUALI: POSSIBILE CHE TU SIA %s
%s ?",nome,cognome);

}
```

Matrici Bidimensionali

Matrici (1)

- Scrivere le seguenti funzioni per la gestione di matrici mxn
  - Prima funzione: una funzione che riempia una matrice di interi mxn direttamente da tastiera

    void StampaMatrice(int A[M][N])
  - Seconda funzione: una funzione che stampi a video una matrice di interi mxn

    void RiempiMatrice(int A[M][N])
  - Terza funzione: una funzione che cerchi il numero di occorrenze di un determinato intero all'intero di una matrice mxn

    int CercaElemento(int ele, int A[M][N])

Matrici (1)

```
void RiempiMatrice(int A[M][N]){
  int i,j;
  for(i=0;i<M;i++){
    for(j=0;j<N;j++)
      {printf("\nInserisci l'elemento A[%d][%d]",i,j);
       scanf("%d",&A[i][j]);
      }
}
  return;
}
```

Matrici (1)

```
void StampaMatrice(int A[M][N])
{
  int i,j;
  for(i=0;i<M;i++){
    for(j=0;j<N;j++)
      printf("A[%d][%d]= %d \n", i,j,A[i][j]);

  }
  return;
}
```

Matrici (1)

```
int CercaElemento(int ele, int A[M][N])
{
  int i,j,ct=0;
  for(i=0;i<M;i++){
    for(j=0;j<N;j++)
      if(A[i][j]==ele)
        ct++;
    }
  return ct;
}
```

Matrici (1)

Scrivere una funzione che data una matrice di interi mxn restituisca l'indice minimo di riga che la somma massima

int RigaMax (int A[M][N])

Matrici (2)

```
int RigaMax (int A[M][N])
{
  int i,j,somma,somma_max=0,n_riga_max=0;
  for(i=0;i<M;i++){
    somma=0;
    for(j=0;j<N;j++)
      somma=somma+A[i][j];
    if(somma > somma_max)
      n_riga_max=i;
     }
  return n_riga_max;
}
```

Le strutture

Typedef

L'istruzione introdotta dalla parola-chiave typedef definisce un sinonimo di un tipo esistente, **cioè non crea un nuovo tipo, ma un nuovo identificatore di un tipo (nativo o astratto) precedentemente definito.**

```
double a1[100]; double a2[100]; double a3[100];  ecc...
```

usando typedef la semplificazione é evidente:

```
typedef double a[100];  a a1;  a a2;  a a3;  ecc...
```

Strutture

Come gli array, le strutture sono gruppi di dati; a differenza dagli array, i singoli componenti di una struttura possono essere di tipo diverso.

Esempio di definizione di una struttura:

```
struct anagrafico
{
  char nome[20];
  int anni;
  char indirizzo[30];
} ;
Struct anagrafico cittadino;
```

Viene cosi definita una nuova struttura anagrafico e definita cittadino di tipo struct anagrafico.

Strutture

Le variabili possono anche essere dichiarate tra "}" e ";" di una dichiarazione di struttura; ad esempio:

```
 struct anagrafico
{
  char nome[20];
  int anni;
  char indirizzo[30];
} cittadino;
```

che equivale al precedente esempio di definizione di una nuova variabile strutturata di nome "cittadino".

Strutture

Una struttura puo' essere pre-inizializzata al momento della dichiarazione:

```
struct anagrafico cittadino={"Uzi",30,”Via Roma”};
```

Per accedere ai membri (o campi) di una struttura il C fornisce l'operatore "."

Ad esempio:

```
cittadino.anni=40;
```

Strutture

Anche con le strutture si puo utilizzare typedef. La seguente istruzione crea un nuovo tipo "" che e' di tipo "struct anagrafico" e puo' essere inizializzato come al solito:

```
typedef struct anagrafico

{
char nome[20];
int anni;
char indirizzo[30];
} dati;
```

```
dati cittadino= {"Uzi",30,”Via Roma”};
```

Strutture

Il C permette anche la definizione array di strutture:

```
dati popolazione[1000];
```

che si possono utilizzare nel seguente modo:

```
popolazione[50].anni=5;
```

dove il campo "anni" del record 50 di popolazione assume valore 5;

Esercizio

Il gioco dei PIRATI

Esercizio

Il gioco dei PIRATI

Cosa ci serve:

1) Definire un tipo pirata che ha i seguenti campi:

Nome di tipo string

Anni di tipo int

2) Creare la ciurma dei pirati, cioè un array di pirati con almeno 5 o 6 pirati

3) Creare un array di stringhe che conterrà l'elenco dei pirati morti durante il combattimento

4) Creare l'elenco delle reclute, cioè un array di pirati pronti a sostituire i compagni deceduti

Esercizio

Il gioco dei PIRATI

Che funzioni ci servono:

0) Una funzione che inserisca i pirati nella ciurma

1) Una funzione che visualizza l'elenco dei pirati formanti la ciurma

2) Una funzione che inserisca i morti in una array

3) Una funzione che visualizzi l'elenco dei pirati deceduti

Esercizio

Il gioco dei PIRATI

Che funzioni ci servono:

4) Una funzione che crei una nuova ciurma contenente solo i pirati che non sono morti

5) Una funzione che crei la ciurma delle reclute

6) Una funzione che inserisca le reclute nella ciurma dei pirati solo se anno almeno 15 anni

Esercizio

inserisci il nome del pirata: Pippo

Inserisci quanti anni ha Pippo:23

Vuoi inserire ancora pirati (0 per si)?0

inserisci il nome del pirata: Pluto

Inserisci quanti anni ha Pluto:34

Vuoi inserire ancora pirati (0 per si)?1

La ciurma dei pirati risulta:

n 0 pirata sgangherato : Pippo e ha 23 anni
n 1 pirata sgangherato : Pluto e ha 34 anni
Inserisci il nome del pirata schiattato:Pippo

Vuoi inserire ancora morti (0 per si)?0

Inserisci il nome del pirata schiattato:Paperino

Vuoi inserire ancora morti (0 per si)?1

Esercizio

Elenco degli schiattati:
Pippo ci ha lasciato
Paperino ci ha lasciato

Adesso li elimino dalla ciurma...
Pippo ci ha lasciato...

La ciurma dei pirati risulta:

n 0 pirata sgangherato : Pluto e ha 34 anni
inserisci il nome del pazzo che vuole diventare pirata: minnie

Inserisci quanti anni ha minnie:23

Vuoi inserire ancora reclute (0 per si)?0

inserisci il nome del pazzo che vuole diventare pirata: paperina

Inserisci quanti anni ha paperina:11

Vuoi inserire ancora reclute (0 per si)?1

Esercizio

La ciurma dei reclute risulta:

n 0 recluta sgangherata : minnie e ha 23 anni
n 1 recluta sgangherata : paperina e ha 11 anni

Adesso ti inserisco le reclute di giusti anni...
minnie ha 23 anni, lo prendiamo
paperina e' un fantolino, che se ne vada a casa

La ciurma finale dei pirati risulta:

n 0 pirata sgangherato : Pluto e ha 34 anni
n 1 pirata sgangherato : minnie e ha 23 anni

Consegna

Per la prossima consegna si deve realizzare il gioco del campo minato!

```
$ ./CampoMinato

1:      ########
2:      ########
3:      ########
4:      ########
5:      ########
6:      ########
7:      ########
8:      ########

        12345678

coordinata x fra 1 e 8:
```

```
coordinata x fra 1 e 8: 1
coordinata y fra 1 e 8: 1

1:        1#####
2:      12#####
3:      1######
4:      2######
5:      1######
6:       1 1 1 2##
7:            1##
8:            1##

        12345678

coordinata x fra 1 e 8:
```