

Problemi**Realizzare una applicazione che risolva ciascuno dei seguenti problemi**

1. Scrivere una funzione che ricevendo una stringa *s*, la stampi in ordine inverso, senza fare uso delle funzioni di stringa.
2. Implementare la funzione **strcmp()**, che date due stringhe *s1* ed *s2*, restituisca -1, 0 o 1 a seconda che *s1* preceda, sia uguale o segua *s2*.
3. Implementare mediante puntatori la funzione **int strlen(string *s)**
4. Scrivere un programma che stampi una data stringa *s* in ciascuno dei seguenti modi:
 - a. mediante la funzione di libreria **cout**;
 - b. mediante indicizzazione, considerando la stringa *s* come un vettore;
 - c. mediante puntatori, considerando come puntatore alla stringa *s* l'indirizzo della prima componente (**&s[0]**).
5. Scrivere un programma che legge una stringa *s* e la trasforma in maiuscola e minuscola mediante uso di puntatori.
6. Scrivere una funzione che dati due array di interi *v1[]* e *v2[]*, di dimensioni rispettive *n1* ed *n2*, copi i primi *n* elementi di *v1[]* nei corrispondenti elementi di *v2[]*. Il prototipo sia:
void CopiaArray(int *v1, int n1, int *v2, int n2, int n);
7. Scrivere una funzione che effettui la somma delle righe di una matrice *mat(R*C)* e la memorizzi in un array *v[]* di interi. Il prototipo sia:
void Somma(int *mat, int r, int c, int *v);
8. Scrivere una funzione che, data una stringa *s* ed un carattere *c* restituisca il puntatore alla prima occorrenza di *c* in *s*. Il prototipo sia:
char * CercaPrimo (char *s, char *c);
9. Scrivere, facendo uso di puntatori, l'algoritmo di ricerca di un elemento in un array, mediante il metodo:
 - a. sequenziale;
 - b. con sentinella;
 - c. binaria.
10. Scrivere l'algoritmo di ordinamento di un array, mediante puntatori, secondo la tecnica:
 - a. di inserzione;
 - b. di selezione;
 - c. di scambio.
11. Implementare le funzioni di stringa seguenti mediante puntatori:
 - a. **char *strcpy(char *s1, char *s2)**
 - b. **char *strcat(char *s2, char *s2)**
12. Scrivere un programma che assegni ad un array *cmd[]* di stringhe non dimensionato alcuni comandi di uso comune nel sistema operativo a disposizione. L'utente sceglie da menu il comando da eseguire e il

programma, tramite l'istruzione **int system (char *cmd)**, esegue il comando richiesto.

Esercizi proposti

1. Descrivere l'effetto delle seguenti funzioni, determinando il significato dell'eventuale valore di ritorno:

a. **int f1 (int *p, int len, int v)**

```
{ int j, i=-1;
  for (j=0; j<len; j++)
  { if (v==*(p+j))
    { i=j;
      break;
    }
  }
```

```
  return i;
}
```

b. **void f2 (char *s)**

```
{ char *t=s;
  while(*t)
  { cout<<*t;
    t++;
  }
  t--;
  while(t>=s)
  { cout<<*t;
    t--;
  }
}
```

c. **void f3 (int n, char vett[])**

```
{ int i;
  for(i=7; i>=0; i--)
  { vett[i]=n%2+48;
    n=n/2;
  }
}
```

2. Date le dichiarazioni:

```
char buffer[100];
```

```
char *s;
```

scrivere le istruzioni *equivalenti* alle seguenti:

a. s = buffer;

- b. `buffer[7] = 'c';`
 c. `cout<<"buffer[7] ="<<buffer[7];`
 d. `for (i=0;i<100;i++) buffer[i]='x';`
3. Date le seguenti dichiarazioni:
`int v1[5]={1, 2, 3, 4, 5},`
`v2[3]={10, 20, 30};`
 determinare l'output di ciascuna delle seguenti istruzioni:
- a. `cout<<*v;`
 b. `cout<<v;`
 c. `cout<<*v++;`
4. Date le dichiarazioni seguenti:
`char buffer[5] = {'a', 'b', 'c', 'd', 'e'} ;`
`char *s ;`
 indicare il valore a cui punta `s` via via che vengono eseguite le seguenti istruzioni :
- a. `cout<<&v1[4]-&v1[2];`
 b. `cout<<&v1[4]-&v2[0];`
 c. `cout<<&v1[4]-5;`
 d. `s = buffer;`
 e. `s++;`
 f. `s = s + 2;`
5. Individuare gli errori sintattici e/o logici nei seguenti frammenti di codice e, dopo averli corretti, descriverne l'effetto.
- a. ...
`int *val, val=50;`
`*val = &val;`
`cout<<val;`
 ...
- b. ...
`int *valptr, val=50;`
`valptr = val;`
`cout<<val;`
- c. ...
`float a=5.1, y;`
`int *ptr;`
`ptr=&a;`
`y=*ptr;`
`cout<<y<<endl;`
 ...
- d. ...
`int ptr1_int, ptr2_int, x;`
`x=100;`
`ptr1int=&x;`

```
ptr2int=ptr1int ;  
cout<<ptr1int<<endl;
```

...

e. ...

```
string s, *ps;  
cin>>s;  
x=100;  
ps=&s[0];  
while (*s) cout<<ps<<endl;
```

...

f. ...

```
char s; int i;  
cin>>s;  
while (*s)  
{ s++, i++; }  
cout<<i<<endl;
```

...

g. ...

```
string s, *ps;  
int i;  
cin>>s;  
while (s[i] && s[i]!=' ') i++;  
ps=&s;  
cout<<ps<<endl;
```

...

6. Descrivere l'effetto delle seguenti dichiarazioni.

- a. `char *s[10];`
- b. `int (*s)[10];`
- c. `char *f(char *s);`
- d. `int (*f)(int n);`
- e. `float *f(char c, int n);`
- f. `int (*f(int n))[5];`

