Esercitazioni di Programmazione II

Chiara Petrioli

Ricevimento e info

- Martedi’ dalle 13 alle 14.30
- Via Salaria 113, terzo piano, stanza 311
- Homework: 3-4 ogni due settimane
- Primo homework da consegnare entro il 14 marzo

Esercizi su stringhe e manipolazione di testi

- Una stringa e' un vettore di caratteri che termina con il carattere '\0'
- Il codice ASCII associa ad ogni carattere un intero (al carattere di fine stringa '\0' e' associato il valore 0)

L'insieme dei caratteri ASCII

	0	1	2	3	4	5	6	7	8	9
0	nul	soh	stx	etx	eot	enq	ack	bel	bs	ht
1	nl	vt	ff	cr	so	si	dle	dc1	dc2	dc3
2	dc4	nak	syn	etb	can	em	sub	esc	fs	gs
3	rs	us	sp	!	"	#	$	%	&	'
4	(	)	*	+	,	-	.	/	0	1
5	2	3	4	5	6	7	8	9	:	;
6	<	=	>	?	@	A	B	C	D	E
7	F	G	H	I	J	K	L	M	N	O
8	P	Q	R	S	T	U	V	W	X	Y
9	Z	[	\	]	^	_	`	a	b	c
10	d	e	f	g	h	i	j	k	l	m
11	n	o	p	q	r	s	t	u	v	w
12	x	y	z	{	|	}	~	del		

I numeri a sinistra della tabella rappresentano le cifre più significative del codice del carattere, espresso in notazione decimale (0-127), mentre quelli in cima alla tabella rappresentano le cifre meno significative del codice del carattere. Per esempio, il codice del carattere 'F' è 70, mentre quello di '&' è 38.

Esercizio 1-caratteri

```
int isdigit(int c)
{
return ((c>='0')&&(c<='9'));
}
```

Si scriva una funzione iterativa che, dato un carattere, verifica se e' una cifra

```
int toupper (int c)
{
   if ((c>='a'&&(c<='z'))
         return (c-'a'+'A');
   return c;
}
```

Si scriva una funzione Iterativa che, dato un carattere, restituisce:
-la corrispondente Lettera maiuscola se Il carattere e' una lettera minuscola
-il carattere inalterato altrimenti

Esercizio 2-caratteri

```
void elimina_maiuscole()
{
int c;
while ((c=getchar() != EOF)
{
    if ((c<‘A’)||(c>’Z’))
    putchar(c);
}
}
```

Si scriva una procedura iterativa che, dato un testo, stampa il testo senza le lettere maiuscole

Esercizio 3-caratteri

```
void elimina_spazi_e_inverti  (int max_size)
{
    int c,count,i;
    char str[max_size];
    count=0;
    while (((c=getchar()) != EOF) &&(count<max_size))
    {
       if (c!=‘ ‘)
       {
          str[count]=c;
          count++;
       }
     }
     str[count]=‘\0’;
     for (i=count-1;i>=0;i--)
        printf(“%c”,str[i]);
}
```

Si scriva una procedura Iterativa che, dato un Testo (di dimensione Minore di max_size), stampa il testo invertito e senza gli spazi

Esercizio 4-caratteri ®

```
void Relimina_spazi_e_inverti  ()
{
    int c;
    if ((c=getchar()) != EOF)
    {
        Relimina_spazi_e_inverti();
        if (c!=‘ ‘)
        putchar ( c);
}
}
```

Si scriva una procedura ricorsiva che, dato un testo stampa il testo invertito e senza gli spazi

Esercizio 1-stringhe

```
/*stampa di una stringa */
void printstringa (char *);

main()
{
char str1[1024];
printf("inserisci stringa \n");
scanf("%s",str1);
printstringa(str1);
……….
}

/*Post: stampa il contenuto della stringa*/
void printstringa (char *s)
{
    for (;*s!='\0';s++)
        putchar(*s);
}
```

Si scriva una procedura iterativa che, data una stringa s presa in input ne stampi i caratteri in output

OPPURE:

```
void printstringa (char *s)
{
      int i;
      for (i=0;*s[i]!='\0';i++)
           putchar(s[i]);
}
```

Esercizio 2-stringhe

```
/*stampa di una stringa */
void stringcopy (char *, char *);

main()
{
char str1[1024];
char str2[1024];
printf("inserisci due stringhe \n");
scanf("%s",str1);
scanf("%s",str2);
stringcopy(str1,str2);
……….
}

/*copia la stringa s2 nel vettore s1. */
/*Pre: dim del vettore s1 sufficienti per contenere s2*/
/*Post: s1 contiene la stringa s2*/


void stringcopy(char *s1, char *s2)
{
      for (;*s2 !='\0';s1++,s2++)
                *s1=*s2;
      *s1='\0';
}
```

Si scriva una procedura iterativa che, date due stringhe copi la seconda stringa nella prima

OPPURE:

void stringcopy (char *s1,char *s2)

```
{
      for (;*s1 =*s2;s1++,s2++)
                ;
}
```

Esercizio 3-stringhe

```
/*calcola la lunghezza di una stringa */
int  String_length(char *);

main()
{
char str1[1024];
printf("inserisci una stringa \n");
scanf("%s",str1);
printf("la lunghezza della stringa e' %d \n", String_length(str1));
……….
}

/*calcola la lunghezza della stringa*/

int String_length(char *s1)
{
     int count=0:
      while (*s1 != '\0')
               {
               count++;
               s1++;
                }
      return count;
}
```

Si scriva una funzione iterativa che, data una stringa ne calcoli il numero di caratteri (escluso il carattere di fine stringa)

Esercizio 4-stringhe

```
/*calcola la lunghezza di una stringa */
int  atoi(const char *);

main()
{
char str1[1024];
printf("inserisci una stringa che rappresenta un intero \n");
scanf("%s",str1);
printf("l'intero che corrisponde alla stringa e' %d \n", atoi(str1));
……….
}

/*calcola la lunghezza della stringa*/

int atoi(const char *s1)
{
     int valore=0:
      while (*s1 != '\0')
               {
               valore= (*s1-'0') + valore * 10;
               s1++;
                }
      return valore;
}
```

Si scriva una funzione iterativa che, data una stringa (che rappresenta un numero intero) converta tale stringa nell'intero corrispondente

Esercizio 5-stringhe

```
/*individua la prima occorrenza di un carattere in una stringa*/
int strchr (char *, int )

main()
{
char str1[1024];
int c1;
printf("inserisci stringa \n");
scanf("%s",str1);
printf("inserisci carattere \n");
c1=getchar();
c1=getchar();
If (strchr(str1,c1)==-1)
printf("il carattere inserito NON compare nella stringa \n");
else
printf("il carattere inserito compare in posizione %d \n",
 strchr(str1,c1));
……….
}

/*individua la prima occorrenza del carattere c nella stringa s*/
/*Pre: s stringa*/
/* Post: indice della prima occorrenza di c nella */
/*stringa. Se tale occorrenza non esiste, restituisce -1*/
int strchr (char *s1, int c)
{
        int i;
        for ( i=0 ; ((s1[i]!= '\0')&&(s1[i]!=c)) ; i++)
                ;
        return ((s1[i]==c)?i : (-1));
}
```

Si scriva una funzione iterativa che, data una stringa s presa in input ed un carattere c restituisca l'indice della occorrenza di c nella stringa, -1 se il carattere non compare

Esercizio 6-stringhe

```
/*confronta due stringhe restituendo 0, un valore*/
/*minore di zero o maggiore di zero a seconda che*/
/*le due stringhe siano uguali, la prima minore della seconda*/
/*la prima maggiore della seconda*/
int strcmp (char *, char * )

main()
{
char str1[1024];
char str2[1024];
printf("inserisci due stringhe \n");
scanf("%s",str1);
scanf("%s",str2);
If (strcmp(str1,str2)==0)
printf("le due stringhe inserite sono uguali \n");
If (strcmp(str1,str2)<0)
printf("la prima stringa e' minore della seconda \n");
If (strcmp(str1,str2)>0)
printf("la prima stringa e' maggiore della seconda \n");
……….
}

int strcmp (char *s1, char *s2)
{
       while((*s1==*s2)&&(*s1!='\0'))
       { s1++;s2++; }
      return (*s1-*s2);
}
```

Si scriva una funzione iterativa che confronta due stringhe Restituisce 0, un valore Minore di 0 o un valore Maggiore di 0 qualora la Prima stringa sia Rispettivamente uguale, Minore o maggiore della seconda

Esercizio 7-stringhe

```
/* Post: restituisce 1 se la prima stringa compare come sottostringa della seconda, 0 altrimenti
 * Se la prima stringa e' vuota restituisce 1 (una stringa vuota e' sottostringa di q.siasi stringa
 * Se la seconda stringa e' vuota e la prima no allora restituisce 0*/
int findstr (char *s1, char *s2)
{
	char *temp_s1;
	char *temp_s2;
	while (*s2!='\0')
	{
		temp_s2=s2;
		temp_s1=s1;
		while ((*s1 == *s2)&&(*s1!='\0')&&(*s2!='\0'))
		{
			s1++;
			s2++;
		}
		if (*s1=='\0')
			return 1;
		else if (*s2=='\0')
			return 0;
		else
		{
			s2=++temp_s2;
			s1=temp_s1;
		}

	}
	return (*s1=='\0');
}
```

Si scriva una funzione Iterativa che, date due stringhe, verifica se la prima stringa e' una sottostringa della seconda

Esercizio 8-stringhe

```
char *strcat (char *s1, const char *s2)
{
    char *temp=s1;
    while (*s1!='\0')
          s1++;
      while (*s1=*s2)
    {
          s1++;
          s2++;
    }
    return temp;
}
```

Si scriva una funzione Iterativa che, date due stringhe, accoda la Seconda stringa alla Prima. Restituisce Il puntatore alla Stringa concatenata

Esercizio 9-stringhe

```
char *inverti_stringa (char *s1)
{
      char *temp;
      char *temp1;
      temp=s1;
      temp1=s1;
      char tempo;

       while (*temp!='\0')
            temp++;
       temp--;
       while (temp1<temp)
       {
                  temp=*temp1;
                   *temp1=*temp;
                   *temp=tempo;
                    temp1++;
                    temp--;
       }
       return s1;
}
```

Si scriva una funzione Iterativa che prende in input una Stringa e Restituisce il puntatore Alla stringa invertita

Esercizio 1-stringhe ®

```
/*stampa di una stringa (ricorsiva)*/
void Rprintstringa (char *);

main()
{
char str1[1024];
printf(“inserisci stringa \n”);
scanf(“%s”,str1);
Rprintstringa(str1);
……….
}

/*Post: stampa il contenuto della stringa*/
void Rprintstringa (char *s)
{
    if (*s != '\0')
            {
            putchar(*s);
            Rprintstringa (s+1);
           }
}
```

Si scriva una procedura ricorsiva che, data una stringa s presa in input ne stampi i caratteri in output

Esercizio 2-stringhe ®

```
/*stampa di una stringa (ricorsiva)*/
void Rprintstringa_invertita (char *);

main()
{
char str1[1024];
printf("inserisci stringa \n");
scanf("%s",str1);
Rprintstringa_invertita(str1);
……….
}
```

Si scriva una procedura ricorsiva che, data una stringa s presa in input ne stampi i caratteri in output in ordine invertito

```
/*Post: stampa il contenuto della stringa invertito*/
void Rprintstringa_invertita (char *s)
{
    if (*s != '\0')
            {
            Rprintstringa_invertita(s+1);
              putchar(*s);
            }
}
```

Esercizio 3-stringhe ®

```
/*calcola il numero di caratteri in una stringa*/
int Rsizestringa (char *, int);

main()
{
char str1[1024];
printf("inserisci stringa \n");
scanf("%s",str1);
printf("il numero di caratteri della stringa e' %d \n",
Rsizestringa(str1,0) );
……….
}
```

Si scriva una procedura ricorsiva che, data una stringa s presa in input calcoli il numero di caratteri della stringa (escluso il carattere di fine stringa)

```
/*calcola il numero di caratteri in una stringa*/
/*Pre: s stringa, i>=0, chiamato con i==0 */
/*Post: restituisce il numero di caratteri della
stringa (escluso il carattere di fine stringa */
int Rsizestringa(char s[ ], int i)
{
    if(s[i]=='\0')
        return 0;
    else
        return (1+Rsizestringa(s,i+1));
}
```

OPPURE:

```
int Rsizestringa (char *s)
{
    if (*s == '\0')
        return 0;
    else
        return (1+Rsizestringa(s+1));
}
```

Esercizio 4-stringhe ®

```
/*individua la prima occorrenza di un carattere in una stringa*/
char * Rstringchr (char *, int )

main()
{
char str1[1024];
int c1;
printf("inserisci stringa \n");
scanf("%s",str1);
printf("inserisci carattere \n");
c1=getchar();
c1=getchar();
If (Rstringchr(str1,c1)==NULL)
printf("il carattere inserito NON compare nella stringa \n");
else
printf("il carattere inserito compare in posizione %d \n",
 (Rstringchr(str1,c1)-str1));
……….
}

/*individua la prima occorrenza del carattere c nella stringa s*/
/*Pre: s stringa*/
/* Post: puntatore alla prima occorrenza di c nella */
/*stringa. Se tale occorrenza non esiste, restituisce NULL*/
char * Rstringchr (char *s, int c)
{
      if(*s=='\0')
            return (NULL);
      else if (*s == c)
            return s;
      else
            return (Rstringchr(s+1,c));
```

Si scriva una procedura ricorsiva che, data una stringa s presa in input ed un carattere c restituisca il puntatore alla prima occorrenza di c in s, NULL se il carattere non compare nella stringa

Esercizio 5-stringhe ®

```
/*individua la prima occorrenza di un carattere in una stringa*/
char * Rstringlastchr (char *, int )
main()
{
char str1[1024];
int c1;
printf("inserisci stringa \n");
scanf("%s",str1);
printf("inserisci carattere \n");
c1=getchar();
c1=getchar();
If (Rstringlastchr(str1,c1)==NULL)
printf("il carattere inserito NON compare nella stringa \n");
else
printf("l'ultima occorrenza del carattere inserito compare in posizione %d \n",
 (Rstringlastchr(str1,c1)-str1));
}
/*Post: restituisce il puntatore all'ultima occorrenza di c nella */
/*stringa, NULL se il carattere non compare*/
char *Rstringlastchr (char *s, int c)
{
	char *temp;
	if (*s=='\0')
		return (NULL);
	else
	{
		temp=Rstringlastchr(s+1,c);
		if(*s==c)
			return((temp!=NULL)?temp:s);
		else
			return temp;
	}
}
```

Si scriva una procedura ricorsiva che, data una stringa s presa in input ed un carattere c restituisca il puntatore All'ultima occorrenza di c in s, NULL se il carattere non compare nella stringa

Homework 1

-Si scriva una funzione ***ricorsiva*** che date due stringhe s1 e s2 calcoli e restituisca un intero contenente la lunghezza del prefisso comune piu' lungo

-Si scriva una funzione ***ricorsiva*** che data una stringa verifichi che sia composta di sole cifre

- Si scriva una funzione *ricorsiva* che data una stringa calcoli il numero di lettere maiuscole nella stringa