

Come verificare se un numero è palindromo in C

SALA MUSIK



Problema

- Inserito un numero in input, stabilire se è palindromo.

Problema



- Inserito un numero in input, stabilire se è palindromo.
 - Un numero è palidromo quando può essere letto sia da destra, sia da sinistra. Es.: 55, 1221.

Progettazione

- Per risolvere il problema presentato, seguirò i seguenti passaggi:



Progettazione



- Per risolvere il problema presentato, seguirò i seguenti passaggi:
 - 1) Farò inserire il numero in input tramite la funzione `inserire()`;

Progettazione

- Per risolvere il problema presentato, seguirò i seguenti passaggi:
 - 1) Farò inserire il numero in input tramite la funzione `inserire()`;
 - 2) Genererò il reverse del numero inserito tramite la funzione `reverseInt()`;

Progettazione



- Per risolvere il problema presentato, seguirò i seguenti passaggi:
 - 1) Farò inserire il numero in input tramite la funzione `inserire()`;
 - 2) Genererò il reverse del numero inserito tramite la funzione `reverseInt()`;
 - 3) Confronterò il numero generato con quello inserito in input. Se sono uguali → il numero è palindromo

Funzione `inserire()`

```
int inserire(void){  
    int N;  
  
    printf("Insert a number: ");  
    scanf("%d", &N);  
    fflush(stdin);  
  
    return N;  
}
```

Consente l'input del numero da controllare

Funzione reverseInt()

- Per capovolgere un numero, adopererò solamente variabili e operatori matematici.

Funzione reverseInt()

- Per capovolgere un numero, adopererò solamente variabili e operatori matematici.
- Le variabili che userò sono:

Funzione reverseInt()

- Per capovolgere un numero, adopererò solamente variabili e operatori matematici.
- Le variabili che userò sono:
 - i

Funzione reverseInt()

- Per capovolgere un numero, adopererò solamente variabili e operatori matematici.
- Le variabili che userò sono:

- i 

Per dividere e moltiplicare iterativamente

Funzione reverseInt()



- Per capovolgere un numero, adopererò solamente variabili e operatori matematici.
- Le variabili che userò sono:
 - i;  Per dividere e moltiplicare iterativamente
 - reversed

Funzione reverseInt()



- Per capovolgere un numero, adopererò solamente variabili e operatori matematici.
- Le variabili che userò sono:
 - i;  Per dividere e moltiplicare iterativamente
 - reversed  Per salvare in memoria il numero rovesciato

Funzione reverseInt()



- Per capovolgere un numero, adopererò solamente variabili e operatori matematici.
- Le variabili che userò sono:
 - i;  Per dividere e moltiplicare iterativamente
 - reversed;  Per salvare in memoria il numero rovesciato
 - seg

Funzione reverseInt()



- Per capovolgere un numero, adopererò solamente variabili e operatori matematici.

- Le variabili che userò sono:

- i; 
- reversed; 
- seg 

Per dividere e moltiplicare iterativamente

Per salvare in memoria il numero rovesciato

Per estrarre parte del numero inserito

Funzione reverseInt()



- Per capovolgere un numero, adopererò solamente variabili e operatori matematici.

- Le variabili che userò sono:

- i; 
- reversed; 
- seg; 
- N

Per dividere e moltiplicare iterativamente

Per salvare in memoria il numero rovesciato

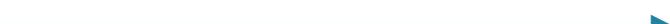
Per estrarre parte del numero inserito

Funzione reverseInt()



- Per capovolgere un numero, adopererò solamente variabili e operatori matematici.

- Le variabili che userò sono:

- i; 
- reversed; 
- seg; 
- N 

Per dividere e moltiplicare iterativamente

Per salvare in memoria il numero rovesciato

Per estrarre parte del numero inserito

Per salvare in memoria il numero inserito

Progetto funzione reverseInt()

- Qual è il piano? Esempio di funzionamento:
 - 1) Isolo le ultime cifre di N nella variabile seg (facendo $N\%i$)

N	i	seg	reversed
456	10	6	0

Progetto funzione reverseInt()

- Qual è il piano? Esempio di funzionamento:
 - 2) Divido seg per $i/10$ così isolo la prima cifra a partire da sinistra.

N	i	seg	reversed
456	10	6	0

Progetto funzione reverseInt()

- Qual è il piano? Esempio di funzionamento:
 - 3) Moltiplico reversed * 10 e aggiungo seg.

N	i	seg	reversed
456	10	6	6

Progetto funzione reverseInt()

- Qual è il piano? Esempio di funzionamento:
3) Moltiplico $i*10$

N	i	seg	reversed
456	100	6	6

Progetto funzione reverseInt()

- Qual è il piano? Esempio di funzionamento:
 - 4) Ripeto

Progetto funzione reverseInt()

- Qual è il piano? Esempio di funzionamento:
 - 5) Isolo le ultime cifre di N nella variabile seg (facendo $N\%i$)

N	i	seg	reversed
456	100	56	6

Progetto funzione reverseInt()

- Qual è il piano? Esempio di funzionamento:
 - 6) Divido seg per $i/10$ così isolo la prima cifra a partire da sinistra.

N	i	seg	reversed
456	100	5	6

Progetto funzione reverseInt()

- Qual è il piano? Esempio di funzionamento:
7) Moltiplico reversed * 10 e aggiungo seg.

N	i	seg	reversed
456	100	5	65

Progetto funzione reverseInt()

- Qual è il piano? Esempio di funzionamento:
8) Moltiplico $i*10$

N	i	seg	reversed
456	1000	5	65

Progetto funzione reverseInt()

- Qual è il piano? Esempio di funzionamento:
9) Ripeto.

Progetto funzione reverseInt()

- Qual è il piano? Esempio di funzionamento:
10) Isolo le ultime cifre di N nella variabile seg (facendo $N\%i$)

N	i	seg	reversed
456	1000	456	65

Progetto funzione reverseInt()

- Qual è il piano? Esempio di funzionamento:
11) Divido `seg` per `i/10` così isolo la prima cifra a partire da sinistra.

N	i	seg	reversed
456	1000	4	65

Progetto funzione reverseInt()

- Qual è il piano? Esempio di funzionamento:
12) Moltiplico reversed * 10 e aggiungo seg.

N	i	seg	reversed
456	1000	4	654

Progetto funzione reverseInt()

- Qual è il piano? Esempio di funzionamento:
13) Moltiplico $i*10$

N	i	seg	reversed
456	10000	4	654

Progetto funzione reverseInt()

- Qual è il piano? Esempio di funzionamento:

14)NON ripeto:

- Perché i è uguale a 10000, che viola la condizione del ciclo while.
- Pertanto, smetto di iterare e la funzione termina e ritorna il valore della variabile reversed.

```
while(i<(N*10)) { ... }  
  
return reversed;
```



Come metto in pratica tutto ciò?



Funzione reverseInt()

- Dichiaro la funzione e le variabili che utilizzerò.

```
int reverseInt(int N){  
    int i=10, reversed=0, seg;  
  
    return reversed;  
}
```

Funzione reverseInt() (2)

- Inizializzo un ciclo while con condizione: $i < (N * 10)$

```
int reverseInt(int N){  
    int i=10, reversed=0, seg;  
  
    while(i < (N*10)){  
  
    }  
  
    return reversed;  
}
```

Funzione reverseInt() (3)

- Scrivo l'algoritmo da iterare all'interno del ciclo while.

```
int reverseInt(int N){
    int i=10, reversed=0, seg;

    while(i<(N*10)){
        seg = N%i;
        seg = (int) seg / (i / 10);
        reversed = (reversed*10)+seg;
        i*=10;
    }

    return reversed;
}
```

Funzione main()

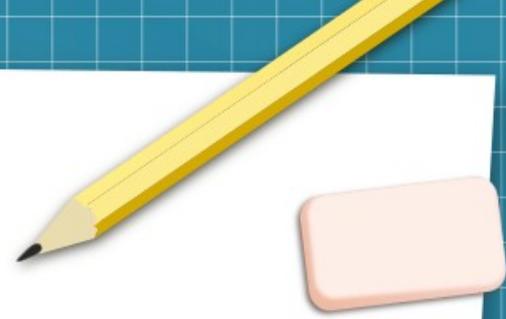
- Nella funzione principale dichiaro le due variabili da comparare.
 - input
 - reversed

```
int main(void){  
    int input, reversed;  
  
    input = inserire();  
    reversed = reverseInt(input);  
  
    if(input==reversed) printf("Numero palindromo");  
    else printf("Inserito numero non palindromo");  
  
    putchar('\n');  
    return 0;  
}
```

Funzione main() (2)

- Nella funzione principale dichiaro le due variabili da comparare.
 - input
 - reversed
- Eseguo il controllo tra le due e stampo l'esito del controllo.

```
int main(void){  
    int input, reversed;  
  
    input = inserire();  
    reversed = reverseInt(input);  
  
    if(input==reversed) printf("Numero palindromo");  
    else printf("Inserito numero non palindromo");  
  
    putchar('\n');  
    return 0;  
}
```



Codice sorgente:

<https://github.com/MatteoBini/palindrome/blob/main/main.c>



SALA MUSIK



This work is licensed under a Creative Commons Attribution-ShareAlike 3.0 Unported License. It makes use of the works of Mateus Machado Luna.

