|  |  |  |  |
| --- | --- | --- | --- |
| **FT\_PRINTF**  int ft\_printf(char \* formato, arg1,...,argn ) | | | **Note** |
| La funzione printf() scrive su stdout gli argomenti passati, eseguendo le opportune conversioni sotto il controllo della stringa di **formattazione format.**  I caratteri ordinari contenuti in format vengono copiati direttamente sullo stdout,  mentre le **direttive di conversione** specificano come devono essere trattati gli argomenti che seguono  Ogni direttiva di conversione inizia con il carattere % e termina con uno dei caratteri **d, i, o, x, X, u, c, s, f, e, E, g, E, p, n, %.** Tra il **carattere di inizio %** e il carattere di conversione possono essere specificati dei **flag**  (**in ordine qualsiasi**), l'ampiezza, la precisione ed un modificatore. | | |  |
| **Struttura della direttiva di conversione**  **%[flags][width][.precision][length]specifier** | | | |
| PARTE BONUS: A SCELTA TRA   * Manage any combination of the following flags: ’-0.’ and the field minimum width under all conversions. * Manage all the following flags: ’# +’ (Yes, one of them is a space).   **NB: Per il 25% di bonus è necessario gestire tutte e 6 le flag** | | | |
| **%** | Obbligatorio | Indica l'inizio di una direttiva di conversione. |  |
| **-** | **Flag opzionale** | **Specifica un incolonnamento a sinistra.** Il valore convertito viene riempito a destra con spazi, piuttosto  che a sinistra con spazi vuoti o zeri.  A - sostituisce uno 0 se sono dati entrambi. |  |
| **0** | **Flag opzionale** | Specifica che nelle conversioni numerI che, i caratteri di riempimento **saranno 0 (zeri) anziché spazi**.  Se compaiono entrambi i flag 0 e -, il flag 0 viene ignorato. Se viene fornita una precisione con una conversione numerica (d, i, o, u, x e X), il flag 0 viene ignorato. Per altre conversioni, il risultato è indefinito. | solo su d,i,u, x,X, C? |
| **'.'** (punto) | Opzionale | Separa l'ampiezza dalla precisione.  Se viene indicato il punto, ma viene omessa la precisione, questa e' assunta a zero. |  |
| **+** | **Flag opzionale** | Specifica che il **segno** verrà sempre stampato. In genere il segno è usato solo per i numeri negativi, il + sovrascrive lo spazio se sono usati entrambi. |  |
| **' ' (spazio)** | **Flag opzionale** | Specifica che verrà stampato uno spazio prima di un numero positivo **.** | solo su d,i,u, x,X, c? |
| **#** | **Flag opzionale** | Specifica un formato alternativo per l'output.  Se il carattere di conversione e' o (ottale) e se il valore da convertire e' diverso da zero, il primo carattere stampato e' 0. Se il carattere di conversione e' x o X (esadecimale) e se il valore da convertire non e' nullo, i primi caratteri stampati saranno ‘0x’ o ‘0X’ a seconda che la direttiva sia x o X. Se il carattere di conversione e' e o E o f o g o G viene sempre stampato il punto decimale. Con g o G gli zeri non significativi non sono rimossi. Per altre conversioni, il risultato è indefinito. |  |
| **Ampiezza** | **Numero opzionale** | **Dimensione** minima del campo.  Se il valore convertito ha meno caratteri rispetto alla larghezza del campo, verrà riempito con spazi a sinistra (o a destra, se è stato assegnato il flag -). L'ampiezza puo' essere sostituita dal carattere **\* o "\*m$"**. In tal caso il valore dell'ampiezza viene letto dal prossimo argomento che deve essere di tipo **int**.  Una larghezza di campo negativa viene considerata come un flag '-' seguito da una larghezza di campo positiva. In nessun caso una larghezza di campo inesistente o piccola causa il troncamento di un campo; se il risultato di una conversione è più largo della larghezza del campo, il campo viene espanso per contenere il risultato della conversione. |  |
| **Precisione** | **Numero opzionale** | Puo' indicare: La **dimensione massima stampabile** della stringa per il carattere di conversione **s**. Il **numero di cifre decimali** per i caratteri di conversione **e**, **E** o **f**. Il **numero di cifre significative** per i caratteri di conversione **g** o **G**. La precisione puo' essere sostituita dal carattere **\***. In tal caso il valore della precisione viene letto dal prossimo argomento che deve essere di tipo **int**. |  |
| **~~Modificatore~~**  **~~h, l~~** ~~e~~ **~~L~~** | **~~Lettera opzionale~~** | ~~La lettera~~**~~h~~**~~seguita da uno dei caratteri di conversione~~**~~d~~**~~,~~**~~i~~**~~,~~**~~u~~**~~,~~**~~o~~**~~,~~**~~x~~**~~o~~**~~X~~**~~indica che l'argomento e' uno~~**~~short int~~**~~o uno~~**~~unsigned short int~~**~~. Il modificatore~~**~~l~~**~~(elle) con gli stessi caratteri di conversione, indica che l'argomento e' un~~**~~long int~~**~~o uno~~**~~unsigned long int~~**~~. Il modificatore~~**~~L~~**~~seguito da uno dei caratteri di conversione~~**~~e~~**~~,~~**~~E~~**~~,~~**~~f~~**~~,~~**~~g~~**~~o~~**~~G~~**~~indica che l'argomento e' un~~**~~long double~~**~~.~~ |  |
| **Caratteri di conversione (Lettera Obbligatoria) (ONLY cspdiuxX%)** | | | |
| **CARATTERE** | **TIPO DI ARG** | **SPECIFICA DI CONVERSIONE** | **NOTE** |
| **d**, **i** | **int** | L'argomento e' convertito a **decimale segnato** |  |
| **u** | **int** | L'argomento e' convertito in **decimale senza segno**. |  |
| **~~o~~** | **~~int~~** | ~~L'argomento e' convertito in~~ **~~notazione ottale~~** ~~e~~ **~~priva di segno~~** | Non previsto |
| **x**, **X** | **int** | L'argomento e' convertito **in notazione esadecimale priva di segno.** A secondo che sia specificato **x** o **X**, le cifre esadecimali impiegate sono rispettivamente **abcdef** o **ABCDEF**. |  |
| **c** | **int** | L'argomento e' convertito in **unsigned char** e viene scritto il carattere risultante. |  |
| **s** | **char \*** | Stampa la **stringa** individuata dall'argomento, fino a raggiungere il carattere '\0'. |  |
| **~~f~~** | **~~double~~** | ~~Converte l'argomento~~ **~~in notazione decimale~~** ~~nel formato [-]ddd.ddd dove le cifre della parte decimale sono date dalla precisione, che per default vale 6.~~ | Non previsto |
| **~~e~~**~~,~~**~~E~~** | **~~double~~** | ~~L'argomento e' stampato~~ **~~in notazione esponenziale~~**~~, in cui viene riportata la mantissa (parte decimale di un numero reale ottenuta facendo la differenza tra il numero e la sua parte intera), la parte decimale e l'esponete preceduto da~~**~~e~~**~~oppure da~~**~~E~~**~~, in funzione del carattere di conversione indicato. La parte decimale e' determinata dalla precisione (default = 6).~~ | Non previsto |
| **~~g~~**~~,~~**~~G~~** | **~~double~~** | ~~L'argomento e' convertito con le regole definite per il carattere di conversione~~**~~f~~**~~o~~**~~e~~**~~o~~**~~E~~**~~.~~ **~~e~~**~~o~~**~~E~~**~~viene utilizzato se l'esponente e' < -4 o e' maggiore o uguale alla precisione. Viene utilizzato~~**~~f~~**~~nell'altro caso.~~ | Non previsto |
| **p** | **void \*** | Viene stampato il valore del **puntatore** passato come argomento. |  |
| **~~n~~** | **~~int \*~~** | ~~Il numero di caratteri scritti vengono~~ **~~memorizzati nell'intero puntato dall'argomento. Nessun argomento viene convertito~~**~~.~~ | Non previsto |
| **%** |  | Stampa il carattere '%'. **Nessun argomento viene trattato.** |  |

|  |  |
| --- | --- |
| **SUBJECT**  **int ft\_printf(const char \*, ...);** | |
| **Don’t implement the buffer management of the original printf().** | Gli output di un programma vengono memorizzati in un buffer e quando si chiama printf vengono estratti. il metodo **setbuf()** viene utilizzato per gestire quel buffer. Quando aggiungi **setbuf(stdout , NULL),** questo dice al buffer che non memorizza gli output e invia direttamente gli output del programma nello stdout. |
| **Your function has to handle the following conversions: cspdiuxX%** | VEDI CARATTERI DI CONVERSIONE |
| **You must use the command ar to create your library. Using the libtool command is forbidden.** |  |
| **Your libftprintf.a has to be created at the root of your repository** |  |
| **Your functions should not quit unexpectedly (segmentation fault, bus error, double free, etc) apart from undefined behaviors.** |  |
| All heap allocated memory space must be properly freed when necessary. No leaks will be tolerated. |  |
| If your project allows you to use your libft, you must copy its sources and its associated Makefile in a libft folder with its associated Makefile. Your project’s Makefile must compile the library by using its Makefile, then compile the project. |  |

|  |  |  |
| --- | --- | --- |
| **PROGETTO** | | |
| **1** | COMPILARE LA LIBRERIA | #include "ft\_printf.h"  #include "ft\_libft.h" |
| **2** | **COSTRUIRE DELLE STRUTTURE PER LE FLAG E LE VARIABILI** | in base alla variabile richiamo la flag |
| **3** | **IMPLEMENTARE FUNZIONI DI COPIA E DI SCRITTURA** | va\_list? |
| **4** | **IMPLEMENTARE PRINTF** |  |
|  | Tester: | <https://github.com/Tripouille/printfTester> |

|  |  |
| --- | --- |
| **FILES** | |
| Program name | libftprintf.a |
| Turn in files | Makefile, \*.h, \*/\*.h, \*.c, \*/\*.c |
| Makefile | NAME, all, clean, fclean, re |
| External functs. | malloc, free, write, va\_start, va\_arg, va\_copy, va\_end |
| Libft authorized | Yes |
| Description | Write a library that contains ft\_printf(), a function that will mimic the original printf() |

va\_list;