

UNIVERSITÀ DEGLI STUDI DI PARMA

Dipartimento di Ingegneria e architettura

Corso di Laurea in Ingegneria Informatica elettronica e delle telecomunicazioni

PROVA PRATICA DI INFORMATICA & LAB. PROGRAMMAZIONE

4 aprile 2024

Nome: _____ Cognome: _____ Matr: _____ Postazione_____

Scrivere un programma in linguaggio C (chiamare il progetto con la propria **<matricola>**) che abbia il comportamento descritto nel seguito. Il tempo a disposizione è di **120 minuti**. Al termine del tempo gli elaborati salvati su **U:** verranno raccolti automaticamente dal sistema di laboratorio. Eventuali documenti sono disponibili in **T:\Bertozzi**, si consiglia di usare **wordpad** per leggere i file di testo.

Si vuole sviluppare un programma in C per calcolare tutti i numeri primi in un intervallo $[1, n]$

A tal fine:

1. definire un array allocato dinamicamente di **unsigned int** inizialmente vuoto
2. chiedere all'utente un numero intero $n > 1$
3. per tutti i numeri x da 1 a n si invochi la funzione **ferm(x)** che restituisce il numero di divisori primi del numero passato per argomento. Se tale funzione restituisce 2 , si incrementi la dimensione dell'array definito al punto #1 e si aggiunga il numero x a tale array
4. si implementi la funzione **ricorsiva unsigned int ferm(unsigned int n)** che svolge i seguenti passi
 - a) se n è uguale a 0 restituisce 0
 - b) se n è uguale a 1 restituisce 1
 - c) se n è pari restituisce $1 + \text{ferm}(n/2)$
 - d) in alternativa si definiscono tre variabili di tipo intero di appoggio: a , b e $b2$
 a viene inizializzata con l'approssimazione per eccesso della radice quadrata di n
 $b2$ viene inizializzata con $a \times a - n$
 b viene inizializzata con la parte intera della radice quadrata di $b2$
fino a che $b2$ non è un quadrato esatto si incrementa a e si ricalcolano i valori di b e $b2$ come indicato
al termine del ciclo se $a + b$ è uguale a n la funzione restituisce 2 , altrimenti restituisce $1 + \text{ferm}(a + b)$
5. Il programma termina stampando quanti numeri primi sono presenti nell'intervallo $[1, n]$ e i relativi valori

Il codice va sviluppato seguendo l'ordine proposto. La correzione termina al primo punto non implementato correttamente.

Al fine di sviluppare il punto #3, inizialmente si definisca uno stub vuoto per la funzione **ferm()** che restituisce un valore prefissato di modo da poter testare il punto #3

Si tenga conto del rischio di overflow per le variabili a , b e soprattutto $b2$

Si guardi sul manuale cosa fa la funzione **double ceil(double x)**