

Fondamenti di Informatica - A.A. 2019-2020

Scuola di Ingegneria Industriale e dell'Informazione
Prof.ssa Cristiana Bolchini
Appello del 01/09/2020



POLITECNICO
MILANO 1863

Cognome

Nome

Matricola o Cod. Persona

Quesito:	1	2	3	4	5	Totale
Valutazione massima:	5	6	6	6	7	30
Valutazione in decimi (/10):						

Istruzioni:

- non è possibile consultare libri, appunti, la calcolatrice o qualsiasi dispositivo elettronico, né comunicare;
- si può scrivere con qualsiasi colore, anche a matita, ad eccezione del rosso.
- tempo a disposizione: 2h

Stile del codice C:

- non è necessario inserire direttive `#include`;
- i commenti non sono necessari, ma potrebbero essere utili nel caso di errore;
- è possibile utilizzare sottoprogrammi di libreria.

Quesito 1 [5 pts]

Dati i due valori $X = 11001110001001_{2C2}$ e $Y = 111100001101_{2MS}$ effettuare la conversione in base 2, notazione complemento a 2 ($2C2$), di ognuno degli operandi sul numero **minimo** di bit necessari. Si effettuino quindi le operazioni $X+Y$ e $X-Y$ indicando esplicitamente se si verifica overflow o meno, e motivando la risposta. Mostrare i passaggi fatti.

Riportare qua la codifica di X_{2C2} , Y_{2C2} e i risultati finali delle operazioni, utilizzando solo le caselle necessarie (**allineati a destra**) ed indicando se si è verificato overflow (segnare la casella corrispondente).

X_{2C2} :

Y_{2C2} :

$(X + Y)_{2C2}$:

$(X - Y)_{2C2}$:

overflow: si ☐ no ☐

overflow: si ☐ no ☐

Quesito 2 [6 pts]

(4 pts) Scrivere un sottoprogramma `compattaarray` che riceve in ingresso un array di numeri interi positivi e qualsiasi altro parametro ritenuto strettamente necessario. Il sottoprogramma rimuove dall'array tutti i numeri non primi e restituisce al chiamante il numero di elementi alla fine presenti nell'array.

(2 pts) Scrivere un programma che chiede all'utente quanti dati vorrà fornire (al più 200) e fino a quando l'utente non dà una risposta valida lo richiede. Acquisiti i valori, il programma chiama il sottoprogramma `compattaarray` e poi visualizza il contenuto dell'array.

Quesito 3 [6 pts]

Un intero $n \geq 0$ si dice "numero narcisistico di ordine x " se la somma delle potenze x -esime delle sue cifre è uguale al numero stesso n . Esempi di numeri narcisistici sono:

- $153 == 1^3 + 5^3 + 3^3$ - narcisista ordine 3
- $1634 == 1^4 + 6^4 + 3^4 + 4^4$ - narcisista ordine 4

Scrivere un sottoprogramma `trovanarcisisti` che, ricevuto in ingresso due array di interi `numero` e `tipo` di egual dimensione, un intero x e qualsiasi altro parametro ritenuto necessario, analizza ciascun valore contenuto in `numero` e se è narcisistico di ordine x mette un 1 nel corrispondente elemento dell'array `tipo`, 0 altrimenti.

Quesito 4 [6 pti]

Una stringa costituisce un pangramma se ogni lettera dell'alfabeto compare almeno una volta. Si scriva un sottoprogramma che ricevuta in ingresso una stringa restituisce 1 se la stringa è un pangramma, 0 altrimenti. La stringa può contenere caratteri maiuscoli, minuscoli e spazi; non importa se il carattere compare come maiuscola o minuscola. Si considerino stringhe di senso compiuto in inglese, in modo tale da non aver il problema delle lettere j, k, w, x, e y, che quindi devono comparire anch'esse.

Esempio:

Ingresso:	A quick brown fox jumps over the lazy dog
Valore restituito:	1
Ingresso:	There is no easy way out of a problem
Valore restituito:	0

Quesito 5 [7 pti]

(1 pto) Definire un tipo di dato opportuno per gli elementi della lista.

(6 pti) Scrivere un sottoprogramma che riceve come parametro una lista di numeri interi `codici`. Il sottoprogramma crea e restituisce una nuova lista `mancanti` contenente tutti i valori interi non presenti in `codici` compresi tra il valore minimo ed il valore massimo di `codici`, ordinati in ordine crescente.

Per esempio, se la lista `codici` contiene i valori

2 -> 5 -> -1 -> -4 -> 3 -|

il sottoprogramma crea e restituisce la seguente lista `mancanti`:

-3 -> -2 -> 0 -> 1 -> 4 -|

Se la lista `codici` contiene i valori

0 -> -5 -> -1 -> -4 -> -3 -> -2 -|

il sottoprogramma crea e restituisce una lista vuota.

Si considerino già disponibili e non da sviluppare i sottoprogrammi seguenti:

```
/* inserisce in testa alla lista */
elem_t * push(elem_t *, int);
/* inserisce in coda alla lista */
elem_t * append(elem_t *, int);
/* inserisce un elemento nella lista in ordine crescente */
elem_t * insert_inc(elem_t *, int);
/* inserisce un elemento nella lista in ordine decrescente */
elem_t * insert_dec(elem_t *, int);
/* elimina dalla lista il primo elemento */
elem_t * pop(elem_t *);
/* elimina dalla lista tutti gli elementi con il valore indicato */
elem_t * delete(elem_t *, int);
/* restituisce il riferimento all'elemento che ha il valore indicato, se esiste, NULL altrimenti */
elem_t * exists(elem_t *, int);
/* restituisce il numero di elementi nella lista */
int length(elem_t *);
```

Sapevo già programmare (rispondere anche se lo si è fatto nel precedente appello):

☐ No, non è vero ☐ in C ☐ in C++/C# ☐ in Python ☐ in Java ☐ in PHP/Javascript ☐ in VB* ☐ in altro linguaggio