

Compito **A** del 10 aprile 2026

Cognome e Nome dello studente: _____

Ha già superato: la 1^a parte la 2^a parte

Macchine logiche

- I/** Data una parola di $n = 4$ bit $\mathbf{a} \doteq (a_3, a_2, a_1, a_0)$, determinare l'espressione algebrica della funzione booleana \mathbf{w} che restituisce, in rappresentazione naturale, il più piccolo indice $i \in \{0, 1, 2, 3\}$ per cui $a_i = 0$. Se questo non esiste, allora la funzione dovrà restituire $i = n$. Esempi: $\mathbf{w}(0, 0, 1, 1) = (0, 1, 0)$, $\mathbf{w}(1, 1, 1, 1) = (1, 0, 0)$. Dare della funzione anche la rappresentazione tabellare e una realizzazione circuitale basata su porte logiche elementari.
- II/** Progettare con la tecnica “parte operativa/parte di controllo” una macchina sincrona che, incorporando il circuito realizzato al punto precedente, fornisca in uscita la sequenza di indici $\mathbf{w}(\mathbf{a})$ per $(\mathbf{a})_2 = 0, 1, \dots, 2^n - 1$ e poi si fermi. Eseguire una simulazione completa del funzionamento della macchina.

Microprocessore e assembler 8086

- III/** Creare due macro, `pushDS` e `popDS`, che consentano di simulare rispettivamente operazioni di push e pop in un array di `byte` allocato nel segmento dati. Lo stack simulato sia pieno e ascendente. Le macro devono avere tre argomenti: il nome del registro usato come sorgente (in caso di push) o destinazione (in caso di pop), il nome dell'array usato come area stack, il nome del registro puntatore usato per indirizzare all'interno dello stack. Esempi di sintassi: `pushDS DL, vect1, SI`; `popDS vect2, AH, BX`.
- IV/** Usare le macro del punto precedente per scrivere un programma ASM8086 che ordini gli M interi positivi (tutti distinti tra loro) di un vettore di input `vect` servendosi di un vettore ausiliario `temp` della stessa dimensione ed inizialmente vuoto. L'algoritmo di ordinamento è il seguente:
1. Svuotare lo stack di input riempiendo quello ausiliario fino a trovare un elemento x in cima al primo che sia inferiore a quello in cima al secondo. Estrarre tale elemento, e porlo in una locazione temporanea;
 2. Effettuare lo svuotamento in senso opposto fino a che x non torni superiore all'elemento in cima allo stack ausiliario. Fare quindi il push di x nello stack ausiliario;
 3. Reiterare i passi precedenti fino a svuotare completamente lo stack di input: lo stack ausiliario conterrà il vettore originale ordinato.

Eseguire una simulazione con il vettore di input 6,7,3,5.

Compito **B** del 10 aprile 2026

Cognome e Nome dello studente: _____

Ha già superato: la 1^a parte la 2^a parte

Macchine logiche

- I/** Data una parola di $n = 4$ bit $\mathbf{a} \doteq (a_3, a_2, a_1, a_0)$, determinare l'espressione algebrica della funzione booleana \mathbf{w} che restituisce, in rappresentazione naturale, il più piccolo indice $i \in \{0, 1, 2, 3\}$ per cui $a_i = 0$. Se questo non esiste, allora la funzione dovrà restituire $i = n$. Esempi: $\mathbf{w}(0, 1, 0, 1) = (0, 0, 1)$, $\mathbf{w}(1, 1, 1, 1) = (1, 0, 0)$. Dare della funzione anche la rappresentazione tabellare e una realizzazione circuitale basata su porte logiche elementari.
- II/** Progettare con la tecnica “parte operativa/parte di controllo” una macchina sincrona che, incorporando il circuito realizzato al punto precedente, fornisca in uscita la sequenza di indici $\mathbf{w}(\mathbf{a})$ per $(\mathbf{a})_2 = 0, 1, \dots, 2^n - 1$ e poi si fermi. Eseguire una simulazione completa del funzionamento della macchina.

Microprocessore e assembler 8086

- III/** Creare due macro, `pushDS` e `popDS`, che consentano di simulare rispettivamente operazioni di push e pop in un array di `byte` allocato nel segmento dati. Lo stack simulato sia vuoto e ascendente. Le macro devono avere tre argomenti: il nome del registro usato come sorgente (in caso di push) o destinazione (in caso di pop), il nome dell'array usato come area stack, il nome del registro puntatore usato per indirizzare all'interno dello stack. Esempi di sintassi: `pushDS DL, vect1, SI`; `popDS vect2, AH, BX`.
- IV/** Usare le macro del punto precedente per scrivere un programma ASM8086 che ordini gli M interi positivi (tutti distinti tra loro) di un vettore di input `vect` servendosi di un vettore ausiliario `temp` della stessa dimensione ed inizialmente vuoto. L'algoritmo di ordinamento è il seguente:
1. Svuotare lo stack di input riempiendo quello ausiliario fino a trovare un elemento x in cima al primo che sia inferiore a quello in cima al secondo. Estrarre tale elemento, e porlo in una locazione temporanea;
 2. Effettuare lo svuotamento in senso opposto fino a che x non torni superiore all'elemento in cima allo stack ausiliario. Fare quindi il push di x nello stack ausiliario;
 3. Reiterare i passi precedenti fino a svuotare completamente lo stack di input: lo stack ausiliario conterrà il vettore originale ordinato.
- Eseguire una simulazione con il vettore di input 8,10,5,7.

Compito C del 10 aprile 2026

Cognome e Nome dello studente: _____

Ha già superato: la 1^a parte la 2^a parte

Macchine logiche

- I/** Data una parola di $n = 4$ bit $\mathbf{a} \doteq (a_3, a_2, a_1, a_0)$, determinare l'espressione algebrica della funzione booleana \mathbf{w} che restituisce, in rappresentazione naturale, il più piccolo indice $i \in \{0, 1, 2, 3\}$ per cui $a_i = 0$. Se questo non esiste, allora la funzione dovrà restituire $i = n$. Esempi: $\mathbf{w}(0, 1, 1, 0) = (0, 0, 0)$, $\mathbf{w}(1, 1, 1, 1) = (1, 0, 0)$. Dare della funzione anche la rappresentazione tabellare e una realizzazione circuitale basata su porte logiche elementari.
- II/** Progettare con la tecnica “parte operativa/parte di controllo” una macchina sincrona che, incorporando il circuito realizzato al punto precedente, fornisca in uscita la sequenza di indici $\mathbf{w}(\mathbf{a})$ per $(\mathbf{a})_2 = 0, 1, \dots, 2^n - 1$ e poi si fermi. Eseguire una simulazione completa del funzionamento della macchina.

Microprocessore e assembler 8086

- III/** Creare due macro, `pushDS` e `popDS`, che consentano di simulare rispettivamente operazioni di push e pop in un array di `byte` allocato nel segmento dati. Lo stack simulato sia pieno e discendente. Le macro devono avere tre argomenti: il nome del registro usato come sorgente (in caso di push) o destinazione (in caso di pop), il nome dell'array usato come area stack, il nome del registro puntatore usato per indirizzare all'interno dello stack. Esempi di sintassi: `pushDS DL, vect1, SI`; `popDS vect2, AH, BX`.
- IV/** Usare le macro del punto precedente per scrivere un programma ASM8086 che ordini gli M interi positivi (tutti distinti tra loro) di un vettore di input `vect` servendosi di un vettore ausiliario `temp` della stessa dimensione ed inizialmente vuoto. L'algoritmo di ordinamento è il seguente:
1. Svuotare lo stack di input riempiendo quello ausiliario fino a trovare un elemento x in cima al primo che sia inferiore a quello in cima al secondo. Estrarre tale elemento, e porlo in una locazione temporanea;
 2. Effettuare lo svuotamento in senso opposto fino a che x non torni superiore all'elemento in cima allo stack ausiliario. Fare quindi il push di x nello stack ausiliario;
 3. Reiterare i passi precedenti fino a svuotare completamente lo stack di input: lo stack ausiliario conterrà il vettore originale ordinato.

Eseguire una simulazione con il vettore di input 10,11,6,7.

Compito **D** del 10 aprile 2026

Cognome e Nome dello studente: _____

Ha già superato: la 1^a parte la 2^a parte

Macchine logiche

- I/** Data una parola di $n = 4$ bit $\mathbf{a} \doteq (a_3, a_2, a_1, a_0)$, determinare l'espressione algebrica della funzione booleana \mathbf{w} che restituisce, in rappresentazione naturale, il più piccolo indice $i \in \{0, 1, 2, 3\}$ per cui $a_i = 0$. Se questo non esiste, allora la funzione dovrà restituire $i = n$. Esempi: $\mathbf{w}(0, 1, 1, 1) = (0, 1, 1)$, $\mathbf{w}(1, 1, 1, 1) = (1, 0, 0)$. Dare della funzione anche la rappresentazione tabellare e una realizzazione circuitale basata su porte logiche elementari.
- II/** Progettare con la tecnica “parte operativa/parte di controllo” una macchina sincrona che, incorporando il circuito realizzato al punto precedente, fornisca in uscita la sequenza di indici $\mathbf{w}(\mathbf{a})$ per $(\mathbf{a})_2 = 0, 1, \dots, 2^n - 1$ e poi si fermi. Eseguire una simulazione completa del funzionamento della macchina.

Microprocessore e assembler 8086

- III/** Creare due macro, `pushDS` e `popDS`, che consentano di simulare rispettivamente operazioni di push e pop in un array di `byte` allocato nel segmento dati. Lo stack simulato sia vuoto e discendente. Le macro devono avere tre argomenti: il nome del registro usato come sorgente (in caso di push) o destinazione (in caso di pop), il nome dell'array usato come area stack, il nome del registro puntatore usato per indirizzare all'interno dello stack. Esempi di sintassi: `pushDS DL, vect1, SI`; `popDS vect2, AH, BX`.
- IV/** Usare le macro del punto precedente per scrivere un programma ASM8086 che ordini gli M interi positivi (tutti distinti tra loro) di un vettore di input `vect` servendosi di un vettore ausiliario `temp` della stessa dimensione ed inizialmente vuoto. L'algoritmo di ordinamento è il seguente:
1. Svuotare lo stack di input riempiendo quello ausiliario fino a trovare un elemento x in cima al primo che sia inferiore a quello in cima al secondo. Estrarre tale elemento, e porlo in una locazione temporanea;
 2. Effettuare lo svuotamento in senso opposto fino a che x non torni superiore all'elemento in cima allo stack ausiliario. Fare quindi il push di x nello stack ausiliario;
 3. Reiterare i passi precedenti fino a svuotare completamente lo stack di input: lo stack ausiliario conterrà il vettore originale ordinato.
- Eseguire una simulazione con il vettore di input 12,15,7,9.