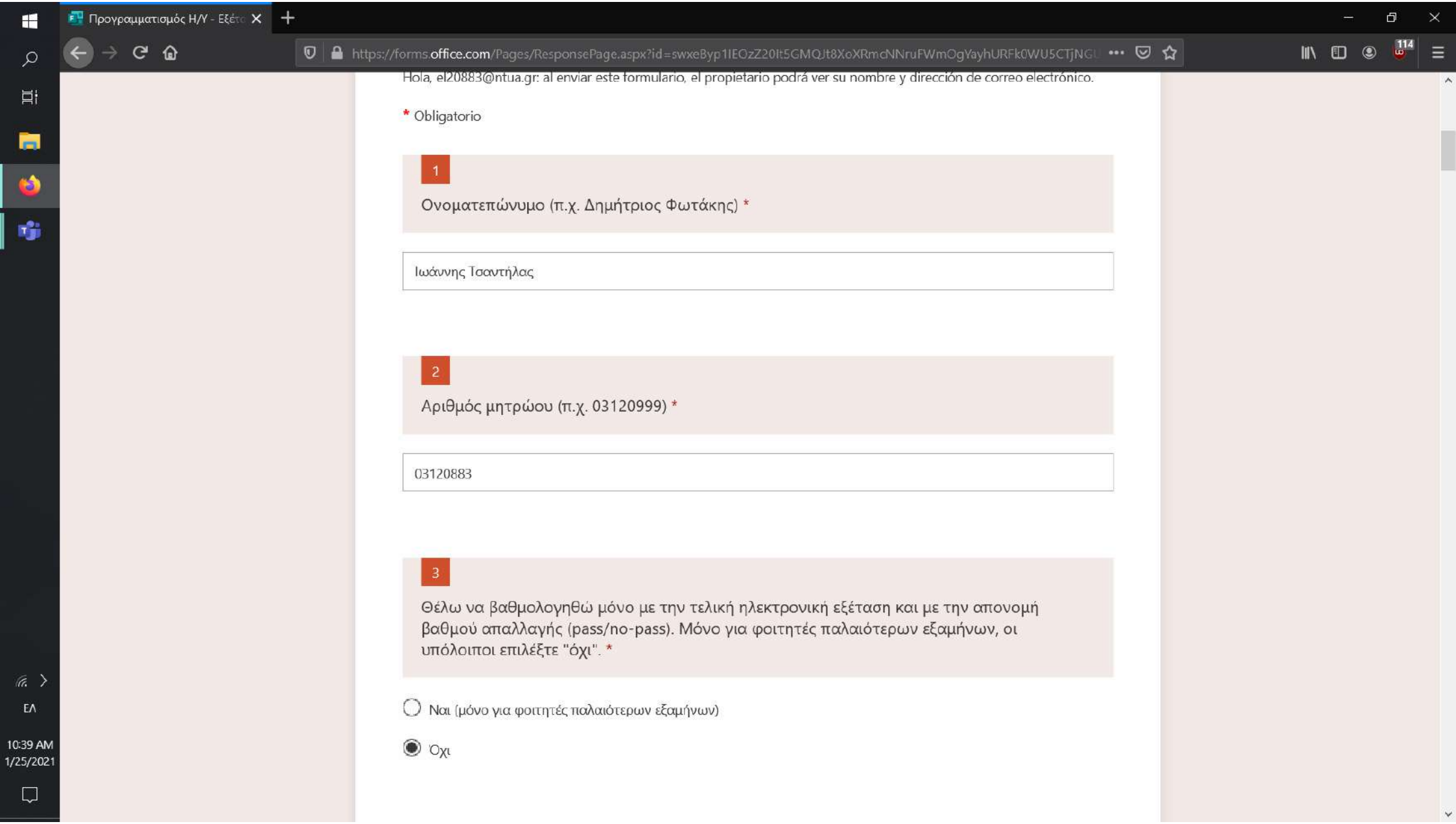
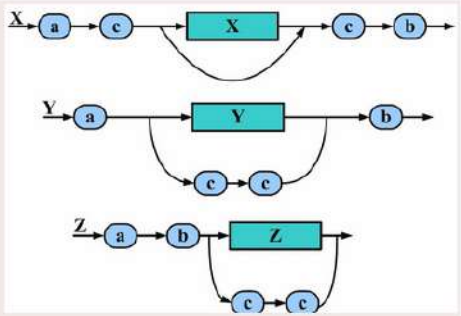

Προγραμματισμός Η/Υ - Εξέτα
https://forms.office.com/Pages/ResponsePage.aspx?id=swxeByp1IEOzZ20It5GMQJt8XoXRmcNNruFWmOgYayhURFk0WU5CTjNG
Hola, el20883@ntua.gr: al enviar este formulario, el propietario podrá ver su nombre y dirección de correo electrónico.
* Obligatorio
1
Ονοματεπώνυμο (π.χ. Δημήτριος Φωτάκης) *
Ιωάννης Τσαντήλας
2
Αριθμός μητρώου (π.χ. 03120999) *
03120883
3
Θέλω να βαθμολογηθώ μόνο με την τελική ηλεκτρονική εξέταση και με την απονομή βαθμού απαλλαγής (pass/no-pass). Μόνο για φοιτητές παλαιότερων εξαμήνων, οι υπόλοιποι επιλέξτε "όχι". *
Ναι (μόνο για φοιτητές παλαιότερων εξαμήνων)
Όχι
ΕΛ
10:39 AM
1/25/2021

4



Ποιο από τα συντακτικά διαγράμματα του σχήματος παράγει τις συμβολοσειρές acb, aaccbb, aaacccbbb και aaaabbbbcccc.

- ○ Το Z
- ◉ Κανένα από τα τρία
- ○ Το X
- ○ Το Y
- ○ Δεν απαντώ

5

Δεδομένου ενός πίνακα μη αρνητικών ακεραίων items μήκους N, ζητείται για κάθε στοιχείο του πίνακα να εκτυπωθεί το τελευταίο (δεξιότερο) από τα προηγούμενα στοιχεία που περιέχει μικρότερη τιμή. Αν δεν υπάρχει κανένα τέτοιο στοιχείο, να εκτυπώνεται -1.

Για παράδειγμα, αν ο πίνακας items περιέχει τα στοιχεία 3, 5, 2, 4, 5, θα πρέπει να εκτυπώνονται κατά σειρά οι αριθμοί -1, 3, -1, 2, 4.

5

Δεδομένου ενός πίνακα μη αρνητικών ακεραίων items μήκους N, ζητείται για κάθε στοιχείο του πίνακα να εκτυπωθεί το τελευταίο (δεξιότερο) από τα προηγούμενα στοιχεία που περιέχει μικρότερη τιμή. Αν δεν υπάρχει κανένα τέτοιο στοιχείο, να εκτυπώνεται -1.

Για παράδειγμα, αν ο πίνακας items περιέχει τα στοιχεία 3, 5, 2, 4, 5, θα πρέπει να εκτυπώνονται κατά σειρά οι αριθμοί -1, 3, -1, 2, 4.

Η παρακάτω συνάρτηση υλοποιεί το ζητούμενο αλλά έχει ένα σημασιολογικό σφάλμα. Βρείτε το και διορθώστε το.

Σαν απάντηση περιμένουμε π.χ., "στη γραμμή 3, x=42".

```
1:  void findSmallerPrevious(int N, int items[]) {
2:    for (int i = 0; i < N; i++) {
3:      int substitute = -1;
4:      for (int j = 0; j < i; j++)
5:        if (items[j] >= items[i]) substitute = items[j];
6:      cout << substitute << endl;
7:    }
8:  }
```

Στην 5η γραμμή του κώδικα βρίσκεται το σφάλμα.

6

Έστω ότι έχετε τρεις διαφορετικούς αλγορίθμους Α, Β και Γ, που επιλύουν το ίδιο πρόβλημα. Η πολυπλοκότητα του Α είναι O((n^3) / (log n)^8), του Β είναι O(n^2 (log n)^5), και του Γ

6

Έστω ότι έχετε τρεις διαφορετικούς αλγορίθμους Α, Β και Γ, που επιλύουν το ίδιο πρόβλημα. Η πολυπλοκότητα του Α είναι O((n^3) / (log n)^8), του Β είναι O(n^2 (log n)^5), και του Γ είναι O(2^n).
Ποιον από τους τρεις θα προτιμούσατε;

- ○ Οποιονδήποτε από τους Β ή Γ, δεν έχουν διαφορά
- ○ Τον Γ
- ○ Τον Β
- ○ Τον Α
- ◉ Δεν απαντώ

7

Ποια είναι η ικανή και αναγκαία συνθήκη ώστε το παρακάτω τμήμα προγράμματος να τερματίζει και να μη συμβεί υπερχείλιση; Εκφράστε την ως μια λογική παράσταση στη C++.

```
while (a != b) a -= 42;
```

a > b && b % 42 == 0

8

8

Στο τέλος της εκτέλεσης του ακόλουθου τμήματος προγράμματος, η τιμή της μεταβλητής t (ως συνάρτηση της τιμής της μεταβλητής n) είναι:

```
t = 0; k = n;
while (k > 0) {
    t += n; k /= 2; }
```

- ○ O(log n)
- ○ O(n^2)
- ○ Δεν απαντώ
- ○ Κανένα από τα τέσσερα
- ○ O(n)
- ◉ O(n log n)

9

Τι επιστρέφει η συνάρτηση search αν κληθεί με low = -50 και up = 40;

9

Τι επιστρέφει η συνάρτηση search αν κληθεί με low = -50 και up = 40;

```
FUNC int f(int x) { return x*x-20*x-1500; }

FUNC int search(int low, int up) {
     int mid=(low+up)/2, val=f(mid);
     if (val == 0) return mid;
     if (val*f(up) <= 0) return search(mid+1, up);
     else return search(low, mid-1); }
```

- ○ Κανένα από τα τρία
- ○ -40
- ○ 50
- ○ Δεν απαντώ
- ◉ -30

10

Τοποθετήστε τους παρακάτω αλγορίθμους σε αύξουσα σειρά (από την μικρότερη πολυπλοκότητα στη μεγαλύτερη) ως προς την χρονική πολυπλοκότητά τους στη χειρότερη περίπτωση:

A. ταξινόμηση με επιλογή (selection sort) ενός πίνακα n στοιχείων
B. ταξινόμηση με συγχώνευση (mergesort) ενός πίνακα n στοιχείων

10

Τοποθετήστε τους παρακάτω αλγορίθμους σε αύξουσα σειρά (από την μικρότερη πολυπλοκότητα στη μεγαλύτερη) ως προς την χρονική πολυπλοκότητά τους στη χειρότερη περίπτωση:

Α. ταξινόμηση με επιλογή (selection sort) ενός πίνακα n στοιχείων
Β. ταξινόμηση με συγχώνευση (mergesort) ενός πίνακα n στοιχείων
Γ. δυαδική αναζήτηση (binary search) σε ταξινομημένο πίνακα n στοιχείων

- ○ Α, Γ, Β
- ○ Β, Γ, Α
- ◉ Γ, Β, Α
- ○ Β, Α, Γ
- ○ Δεν απαντώ

11

Πόσους διαφορετικούς αριθμούς θα εκτυπώσει η ακόλουθη διαδικασία, αν κληθεί με n=40;

```
PROC proc(int n){
    if (n == 0) { WRITELN(n); WRITELN(n+1); }
    else { WRITELN(2*n); proc(n-1); WRITELN(2*n+1); } }
```

- ○ 41
- ○ Δεν απαντώ

○ Δεν απαντώ

11

Πόσους διαφορετικούς αριθμούς θα εκτυπώσει η ακόλουθη διαδικασία, αν κληθεί με n=40;

```
PROC proc(int n){
    if (n == 0) { WRITELN(n); WRITELN(n+1); }
    else { WRITELN(2*n); proc(n-1); WRITELN(2*n+1); } }
```

○ 41

○ Δεν απαντώ

◉ 81

○ 42

○ 82

12

Τι τυπώνει το παρακάτω πρόγραμμα, αν όπου <ΑΜ> θέσετε τον Αριθμό Μητρώου σας;

```
int x;

PROC test(int n, int &k) {
    WRITELN(x, n, k);
    x = x / 3; n += 5; k += n;
    if (n < x) test(k/2, n);
```

12

Τι τυπώνει το παρακάτω πρόγραμμα, αν όπου <ΑΜ> θέσετε τον Αριθμό Μητρώου σας;

```
int x;

PROC test(int n, int &k) {
    WRITELN(x, n, k);
    x = x / 3; n += 5; k += n;
    if (n < x) test(k/2, n);
    else n -= k;
    WRITELN(x, n, k);
}

PROGRAM {
    x = 10 + <AM> % 100; test(x+5, x); WRITELN(x); }
```

93 98 93
134 67 103
44 -103 175
44 175 44
44

13

Έστω μία υλοποίηση δυαδικού δένδρου, οι κόμβοι του οποίου έχουν τον τύπο:

```
struct node {
    int data;
    node *left, *right;
};
```

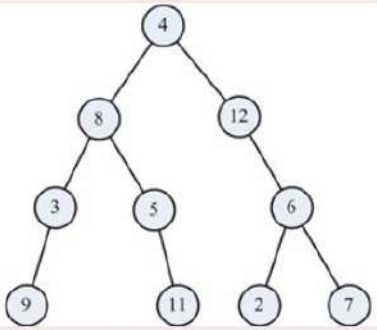
13

Έστω μία υλοποίηση δυαδικού δένδρου, οι κόμβοι του οποίου έχουν τον τύπο:

```
struct node {
    int data;
    node *left, *right;
};
typedef node *tree;
```

Τι τυπώνει η παρακάτω συνάρτηση όταν κληθεί για το δέντρο του σχήματος:

```
int test(tree t) {
    if (t == nullptr) return 0;
    else {
        int k = (t->left != nullptr) +
                (t->right != nullptr);
        if (k <= 1)
            return t->data + test(t->left) +
                   test(t->right);
        else
            return test(t->left) + test(t->right);
    }
}
```



- ○ 38
- ○ 49
- ○ Κανένα από τα τρία
- ◉ Δεν απαντώ
- ○ 67

14

Ο αλγόριθμος γραμμικής αναζήτησης για την εύρεση στοιχείου σε μη ταξινομημένο πίνακα με n στοιχεία:

- ○ Απαιτεί χρόνο O((log n)^2) στη χειρότερη περίπτωση.
- ○ Δεν απαντώ
- ○ Απαιτεί χρόνο O(n^(0.5)) στη χειρότερη περίπτωση.
- ◉ Απαιτεί χρόνο O(n) στη χειρότερη περίπτωση.
- ○ Απαιτεί χρόνο τουλάχιστον λογαριθμικό στο n (δηλ. Ω(log n)) στην καλύτερη περίπτωση.

15

Έστω A ταξινομημένος πίνακας N ακεραίων αριθμών και x ακέραιη μεταβλητή τέτοια ώστε A[0] <= x <= A[N-1]. Ποια λογική πρόταση είναι σωστή αναλλοίωτη για το σημείο «1» του παρακάτω βρόχου;

```
int L = 0, R = N-1;
while (L < R) { /* 1 */
  int M = (L+R)/2, key = A[M];
  if (key <= x) L = M; else R = M-1;
}
```

- ○ L <= R και A[L] <= x < A[R]

15

Έστω Α ταξινομημένος πίνακας Ν ακεραίων αριθμών και x ακέραιη μεταβλητή τέτοια ώστε A[0] <= x <= A[N-1]. Ποια λογική πρόταση είναι σωστή αναλλοίωτη για το σημείο «1» του παρακάτω βρόχου;

```
int L = 0, R = N-1;
while (L < R) { /* 1 */
  int M = (L+R)/2, key = A[M];
  if (key <= x) L = M; else R = M-1;
}
```

- ○ L <= R και A[L] <= x < A[R]
- ◉ L < R και key = A[(L+R)/2]
- ○ Κανένα από τα τρία
- ○ Δεν απαντώ
- ○ L < R και A[L] <= x <= A[R]

16

Τι τυπώνει το παρακάτω πρόγραμμα;

```
PROGRAM {
```

16

Τι τυπώνει το παρακάτω πρόγραμμα;

```
PROGRAM {
     int *p = new int, *q = new int, *t = new int;
     *p=17; *q=3; *t=42;
     p=t; t=q;
     *t=*p**q ; *q=*q+*t;
     WRITELN(*t+*p);
}
```

- ○ Δεν απαντώ
- ○ Κανένα από τα τέσσερα
- ○ 378
- ○ 119
- ○ 210
- ◉ 294

17

Ποια είναι η τιμή της μεταβλητής t στο τέλος της εκτέλεσης του ακόλουθου τμήματος προγράμματος;

```
int n=1, p=1, t=1;
do { n*=4; p*=n; t++; } while (p <= 128000000 AND t <= 8);
```

294

17

Ποια είναι η τιμή της μεταβλητής t στο τέλος της εκτέλεσης του ακόλουθου τμήματος προγράμματος;

```
int n=1, p=1, t=1;
do { n*=4; p*=n; t++; } while (p <= 128000000 AND t <= 8);
```

8

Δεν απαντώ

6

20

7

18

Τι τυπώνει το παρακάτω πρόγραμμα;

```
PROGRAM {
    int *p = new int[20], *q;
    int sum=0, i;
    for (i=1; i<=20; i++) *(p+i-1)=4*i;
    for (i=1, q=p+2; i<=9; i++, q++) sum+=*(p+i/2)+*(q+i);
    WRITELN(sum);
}
```

○ 7

18

Τι τυπώνει το παρακάτω πρόγραμμα;

```
PROGRAM {
    int *p = new int[20], *q;
    int sum=0, i;
    for (i=1; i<=20; i++) *(p+i-1)=4*i;
    for (i=1, q=p+2; i<=9; i++, q++) sum+=*(p+i/2)+*(q+i);
    WRITELN(sum);
}
```

○ 204

○ 356

○ Κανένα από τα τρία, γιατί το 2ο for-loop βγαίνει εκτός ορίων του πίνακα p.

○ Δεν απαντώ

◉ 548

19

Τι επιστρέφει η παρακάτω συνάρτηση αν κληθεί με x = 20 και n = 4096;

```
FUNC int fun(int x, int n) {
    if (n == 0) return x;
```

19

Τι επιστρέφει η παρακάτω συνάρτηση αν κληθεί με x = 20 και n = 4096;

```
FUNC int fun(int x, int n) {
    if (n == 0) return x;
    else return x + fun(x-1, n/2); }
```

- (●) 189
- ( ) 200
- ( ) Δεν απαντώ
- ( ) Κανένα από τα τέσσερα
- ( ) 181
- ( ) 195

20

Σε έναν πίνακα μη αρνητικών ακεραίων items μήκους N, μπορούμε κάθε φορά να αυξάνουμε ένα στοιχείο του κατά 1. Ζητείται να βρεθεί ποιος είναι ο ελάχιστος αριθμός τέτοιων αυξήσεων που πρέπει να κάνουμε, έτσι ώστε τα στοιχεία του πίνακα να είναι σε (γνησίως) αύξουσα σειρά.

Για παράδειγμα, αν ο πίνακας items περιέχει τα στοιχεία 1, 0, 2, ο ελάχιστος αριθμός

20

Σε έναν πίνακα μη αρνητικών ακεραίων items μήκους N, μπορούμε κάθε φορά να αυξάνουμε ένα στοιχείο του κατά 1. Ζητείται να βρεθεί ποιος είναι ο ελάχιστος αριθμός τέτοιων αυξήσεων που πρέπει να κάνουμε, έτσι ώστε τα στοιχεία του πίνακα να είναι σε (γνησίως) αύξουσα σειρά.

Για παράδειγμα, αν ο πίνακας items περιέχει τα στοιχεία 1, 0, 2, ο ελάχιστος αριθμός αυξήσεων είναι 3, έτσι ώστε ο πίνακας τελικά να περιέχει τα στοιχεία 1, 2, 3.

Η παρακάτω συνάρτηση υλοποιεί το ζητούμενο αλλά έχει ένα σημασιολογικό σφάλμα. Βρείτε το και διορθώστε το.

Σαν απάντηση περιμένουμε π.χ., "στη γραμμή 3, x=42".

```
1:  int makeIncreasing(int N, int items[]) {
2:    int result = 0;
3:    for (int i = 1; i < N; i++) {
4:      if (items[i] <= items[i-1]) {
5:        result = items[i-1] - items[i] + 1;
6:        items[i] = items[i-1] + 1;
7:      }
8:    }
9:    return result;
10: }
```

Δεν απαντώ

21

Θα ονομάζουμε έναν θετικό ακέραιο αριθμό «τυχερό» αν τα μοναδικά ψηφία που εμφανίζονται στη δεκαδική του αναπαράσταση είναι το 4 και το 2. Η παρακάτω συνάρτηση ελέγχει αν ένας αριθμός είναι ή όχι τυχερός. Συμπληρώστε ό,τι λείπει στο σημείο που σημειώνεται με ερωτηματικά, προκειμένου να λειτουργεί σωστά.

```
1:  bool isLucky(int n) {
2:    while (/* ??? */) {
3:      int digit = n % 10;
4:      if (digit != 4 && digit != 2) return false;
5:      n /= 10;
6:    }
7:    return true;
8:  }
```

(n/10 != 0) && (n%10 != 0)

22

Σε ποια περίπτωση το παρακάτω πρόγραμμα θα εκτυπώσει την τιμή 171;

```
PROC test(int n, int r) {
    n = n / r; r = n % r; }

PROGRAM {
```

22

Σε ποια περίπτωση το παρακάτω πρόγραμμα θα εκτυπώσει την τιμή 171;

```
PROC test(int n, int r) {
    n = n / r; r = n % r; }
PROGRAM {
    int n = 1717, r = 10;
    test(n, r); test(n, r);
    WRITELN(n+r); }
```

- (●) Αν η επικεφαλίδα της test αλλάξει σε: PROC test(int &n, int &r)
- ( ) Αν η επικεφαλίδα της test αλλάξει σε: PROC test(int n, int &r)
- ( ) Αν η επικεφαλίδα της test αλλάξει σε: PROC test(int &n, int r)
- ( ) Αν η επικεφαλίδα της test δεν αλλάξει.
- ( ) Κανένα από τα τέσσερα
- ( ) Δεν απαντώ

23

Προγραμματισμός Η/Υ - Εξέτα

https://forms.office.com/Pages/ResponsePage.aspx?id=swxeByp1IEOzZ20It5GMQJt8XoXRmcNNruFWmOgYayhURFk0WU5CTjNG

23

Ποια είναι η τιμή της μεταβλητής t στο τέλος της εκτέλεσης του ακόλουθου τμήματος προγράμματος;

```
int n = 10, z = 0, k, t = 0;
FOR (i, 2 TO n) {
   z += n; k = 0;
   FOR (j, 1 TO z) k = k+1;
   t = t + k/10;
}
```

- ○ Δεν απαντώ
- ○ Κανένα από τα τρία
- ○ 66
- ◉ 45
- ○ 55

☑ Enviarme una confirmación por correo electrónico de mis respuestas

Enviar

ΕΛ

10:40 AM
1/25/2021