

Prova de síntesi 2021/22-2

Assignatura	Codi	Data	Hora inici
Pràctiques de programació	05.555	18/6/2022	17:30

O(x)

!!!	Aquesta prova només la poden fer els estudiants que hagin aprovat l'avaluació contínua
------------	---

Aquest enunciat també correspon a les assignatures següents:

- 11.578 - Pràctiques de programació

Fitxa tècnica de la prova de síntesi

- No és necessari que escriguis el teu nom. Un cop resolta la prova final, només s'accepten documents en format .doc, .docx (Word) i .pdf.
- Comprova que el codi i el nom de l'assignatura corresponen a l'assignatura de què t'has matriculat.
- Temps total: 1 **hora** Valor de cada pregunta:
- Es pot consultar cap material durant la prova de síntesi? Quins materials estan permesos?
- Es pot fer servir calculadora? De quin tipus?
- Si hi ha preguntes tipus test, descompten les respostes errònies? Quant?
- Indicacions específiques per a la realització d'aquesta prova de síntesi:

Prova de síntesi 2021/22-2

Assignatura	Codi	Data	Hora inici
Pràctiques de programació	05.555	18/6/2022	17:30

Enunciats

Pregunta 1 [50%]

Continuant amb l'aplicació per a la gestió de la pandèmia de la COVID, en aquest exercici es treballarà només amb els següents tipus:

```
const
    MAX_PERSONS: integer = 100;
    MAX_APPOINTMENTS: integer = 20;
end const

type
    tDate = record
        day : integer;
        month : integer;
        year : integer;
    end record

    tTime = record
        hour : integer;
        minutes : integer;
    end record

    tDateTime = record
        date : tDate;
        time: tTime;
    end record

    tVaccine = record
        name : string;
        required : integer;
        days : integer;
    end record

    tPerson = record
        document : string;
        name : string;
        surname : string;
        email : string;
        address : string;
        cp : string;
        birthday : tDate;
    end record

    tPopulation = record
        elems : vector[MAX_PERSONS] of tPerson;
        count : integer;
    end record
```

Prova de síntesi 2021/22-2

Assignatura	Codi	Data	Hora inici
Pràctiques de programació	05.555	18/6/2022	17:30

```

tAppointment = record
    timestamp : tDateTime;
    person : tPerson;
    vaccine : tVaccine;
end record

tAppointmentData = record
    elems : vector[MAX_APPOINTMENTS] of tAppointment;
    count : integer;
end record
end type

```

Donada la funció:

```

function population_with_vaccine(appointments: tAppointmentData, vaccine: string):
tPopulation

```

que donades unes cites i una vacuna retorna la llista de persones que tenen una cita programada per a aquesta vacuna. Per a simplificar l'exercici podem assumir que la mateixa persona tindrà com a màxim una cita, és a dir, una persona no podrà tenir dues o més cites programades.

Es demana:

- a) **[10%]** Indica les pre-condicions per a aquesta funció. En aquest cas, s'assumeix que la llista de cites no està buida i que existeix almenys alguna persona amb cita per a la vacuna que ens donen. **Justifica la resposta amb una o dues línies de text.**

$\{ \text{appointments} = \text{APPOINTMENTS} \text{ and } \text{APPOINTMENTS.count} > 0 \text{ and } \exists i : 0 < i \leq \text{APPOINTMENTS.count} : \text{APPOINTMENTS.elems}[i].\text{vaccine.name} = \text{vaccine} \}$

La llista de cites no està buida si tenim almenys un element. Ens diu l'enunciat que existeix una cita amb la vacuna, no farà falta comprovar a quina persona està associada

- b) **[40%]** Implementa la funció en **llenguatge algorítmic** aplicant la metodologia de disseny descendent. També és necessari implementar totes les funcions que necessitis crear en els diferents nivells. Per a cada mètode, descriu breument la funció que realitzen (escriu una o dues línies de text per mètode). **No es valoraran mètodes que no tinguin la seva corresponent descripció.**

Prova de síntesi 2021/22-2

Assignatura	Codi	Data	Hora inici
Pràctiques de programació	05.555	18/6/2022	17:30

Nivell 1

```

function population_with_vaccine( appointments: tAppointmentData, vaccine: string):
tPopulation
var
    population: tPopulation;
    i: integer;
end var
    { Initialize variables }
    init_population(population);

    { Loop all the appointments }
    for i:=1 to appointments.count do
        { Filter by vaccine }
        if person_uses_vaccine(appointments.elems[i], vaccine) then
            { Add person to population with searched vaccine }
            add_person(population, appointments.elems[i]);
        end if
    end for

    return population;
end function

```

Nivell 2

```

    { Check if the person uses a vaccine on the appointment }
    function person_uses_vaccine( appointment: tAppointment, vaccine: string): boolean
        return appointment.vaccine.name = vaccine;
    end function

    { Initialize the population table }
    action init_population(inout population: tPopulation)
        population.count := 0;
    end action

    { Add a person to the population table }
    { We assume that it is not necessary to check "duplicate" person }
    action add_person(inout population: tPopulation, in appointment: tAppointment)
        { Update length of the population }
        population.count := population.count + 1;

        { Add person to the population }
        { It also possible to make a copy, not just point to the same person }
        population.elems[population.count] := appointment.person;
    end action

```

Prova de síntesi 2021/22-2

Assignatura	Codi	Data	Hora inici
Pràctiques de programació	05.555	18/6/2022	17:30

Prova de síntesi 2021/22-2

Assignatura	Codi	Data	Hora inici
Pràctiques de programació	05.555	18/6/2022	17:30

Pregunta 2 [50%]

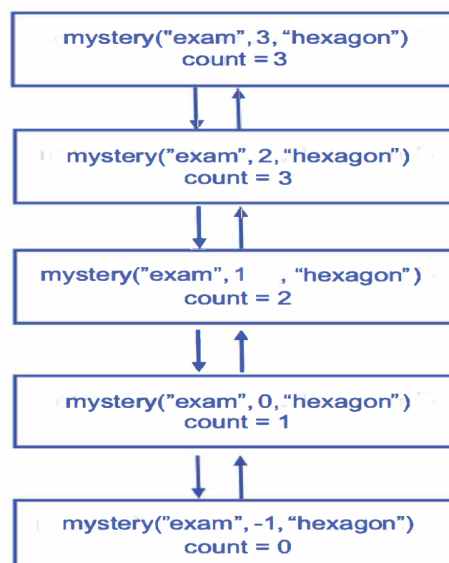
Donada la següent implementació en llenguatge C d'un mètode *mystery*, on la funció:
`int search(char a, char c[N])`
 cerca el caràcter *a* a la cadena *c*, retornant 1 si el troba, 0 en cas contrari:

```
#define N 128
int mystery (char p[N], int x, char q[N])
{
    assert (-1 <= x && x < N);
    int count;
    if (x < 0)                                (1)
        count = 0;                            (2)
    else                                       (3)
        count = mystery (p, x-1, q) + search(p[x], q); (4)
    return count;                             (5)
}
```

- a) [10%] Explica què fa aquest mètode *mystery*, és a dir, quina funció fa. També mostra pas a pas (incloent-hi un diagrama de caixes per a seqüència de crides i els seus paràmetres) el funcionament de *mystery* per a la següent invocació:

```
int ret;
ret = mystery("exam", 3, "hexagon");
```

Aquest mètode retorna el nombre de caràcters dels primers **x+1** caràcters de la cadena **p** que són presents a la cadena **q**.



Prova de síntesi 2021/22-2

Assignatura	Codi	Data	Hora inici
Pràctiques de programació	05.555	18/6/2022	17:30

b) [5%] Explica quines conseqüències tindria substituir la línia (4) per aquesta línia:

count = mystery (p, x, q) + search(p[x], q);

Si el valor del paràmetre **x** és inicialment més gran que -1, llavors mai acabarà la recursió ja que la variable **x** mai canvia de valor en crides recursives successives.

c) [20%] Transforma aquest mètode recursiu en un mètode iteratiu en llenguatge C. Descriu breument la implementació adoptada (en cas contrari, no s'avaluarà l'exercici).

```
#define N 128
int mystery (char p[N], int x, char q[N])
{
    assert (-1 <= x && x < N);
    count = 0;
    while (x >= 0) {
        count = count + search(p[x], q);
        x = x-1;
    }
    return count;
}
```

El mètode recorre a cada iteració del bucle una posició del vector **p** (començant a la posició **x**) analitzant a través de la funció **search** si aquest element es troba al vector **q**. A cada iteració es decrementa **x** en una posició. El bucle finalitza quan la posició **x** arriba al valor 0.

d) [15%] Analitza la complexitat d'aquest mètode (en la versió recursiva) assumint que la funció de temps de **search(a, c)** és $T_{\text{search}} = s$, on **s** és la longitud de **c** en caràcters.

Per analitzar la complexitat computacional d'aquesta funció, cal observar que a cada crida recursiva el valor del paràmetre **x** es decrementa en una unitat fins que arriba al valor -1 on finalitza la recursivitat. Per tant, la funció de temps es defineix en funció del paràmetre **x**.

El cas base (línies 1 i 2) es limita a un condicional i una assignació. En tractar-se de dues operacions elementals se s'agrupa el cost en una única constant k_1 .

Per calcular el cas recursiu, partim d'un condicional (línia 1), una assignació, una operació aritmètica i un accés a vector (línia 4) amb costos constants agrupats en k_2 , als quals hem de sumar el cost de la crida recursiva i de la crida a la funció **search**. Per al càlcul de la crida recursiva tindrem en compte el pitjor cas possible, que és que la cadena **q** tingui sempre **N** caràcters, per la qual cosa el temps serà $T_{\text{search}} = N$.

Podem definir la funció de temps de la següent manera:

$$\begin{aligned} T(x) &= k_1, & \text{si } x &= -1 \\ T(x) &= k_2 + T(x-1) + N, & \text{si } x &\geq 0 \end{aligned}$$

Prova de síntesi 2021/22-2

Assignatura	Codi	Data	Hora inici
Pràctiques de programació	05.555	18/6/2022	17:30

Es resol la recurrència repetint la substitució recursiva un total de i vegades:

$$\begin{aligned}
 T(x) &= k_2 + T(x-1) + N = \\
 &= k_2 + (k_2 + T((x-1)-1) + N) + N = 2 \cdot k_2 + T(x-2) + 2 \cdot N = \\
 &= 2 \cdot k_2 + (k_2 + T((x-2)-1) + N) + 2 \cdot N = 3 \cdot k_2 + T(x-3) + 3 \cdot N = \\
 &= 3 \cdot k_2 + (k_2 + T((x-3)-1) + N) + 3 \cdot N = 4 \cdot k_2 + T(x-4) + 4 \cdot N = \\
 &= \dots = \\
 &= i \cdot k_2 + T(x-i) + i \cdot N
 \end{aligned}$$

Calculem el valor de la variable i que fa que es pugui fer servir la part no recurrent de la definició de $T(x)$:

$$\begin{aligned}
 x - i &= -1 \\
 x + 1 &= i
 \end{aligned}$$

Finalitzem el càlcul de $T(x)$ amb el valor calculat per la variable i :

$$\begin{aligned}
 T(x) &= i \cdot k_2 + T(x-i) + i \cdot N = (x+1) \cdot k_2 + T(x-x-1) + (x+1) \cdot N = \\
 &= x \cdot k_2 + k_2 + T(-1) + x \cdot N + N = x \cdot k_2 + k_2 + k_1 + x \cdot N + N = \\
 &= x \cdot (k_2 + N) + k_2 + k_1 + N
 \end{aligned}$$

La conclusió és que la funció té una complexitat lineal **O(x)**.

Prova de síntesi 2021/22-2

Assignatura	Codi	Data	Hora inici
Pràctiques de programació	05.555	18/6/2022	17:30