

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:

- [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.**
- [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

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");
```

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

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

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

- 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.