
Test C

Naam :

Verzorg je bladspiegel. Volg strikt het gevraagde en implementeer zo efficiënt mogelijk. Alle vragen beantwoord je in C. Je hoeft nergens includes te schrijven. Je mag bibliotheekfuncties gebruiken.

Enkel de functie realloc mag nergens gebruikt worden.

1) Gegeven onderstaande definitie van een gelinkte lijst, alsook van de structs bestand en datum.

<pre>typedef struct { int dag; char maand[30]; int jaar; } datum; typedef struct knoop knoop; struct knoop { bestand b; knoop *next; };</pre>	<pre>typedef struct { char naam[20]; double grootte; datum datum_van_aanmaak; } bestand;</pre>
---	---

Schrijf eerst een logische functie **controleer_bestand(b,g)** die nagaat of een bestand **b** een grootte **g** heeft en aangemaakt werd in augustus 2019.

Schrijf nu een functie ***verwijder_knopen(l,g)*** die uit de gelinkte lijst *l* alle knopen verwijdert waarvan het bestand een grootte *g* heeft en dat werd aangemaakt in augustus 2019. Om bestanden te controleren op datum van aanmaak en op grootte gebruik je vanzelfsprekend de functie ***controleer_bestand(b,g)***. De functie geeft een gelinkte lijst met de verwijderde knopen als resultaat terug. De volgorde waarin deze knopen staan is niet van belang.

- 2) Een vaak voorkomend probleem is dat je in een array wil weten wat het grootste verschil is tussen twee opeenvolgende elementen en meer specifiek welke twee opeenvolgende elementen dat zijn.

Gegeven onderstaand hoofdprogramma:

```
int main(){
    //voor array van ints
    int int_array[]={1,2,3,7,-2};
    int *loc=bepaal_grootste_vershil(...);
    printf("%d->%d\n",loc[0],loc[1]);        // 7 -> -2
    return 0;
}
```

De algemene functie **bepaal_grootste_vershil(...)** bepaalt het grootste verschil tussen twee opeenvolgende elementen van een array van eender welk type en geeft het adres van het eerste van deze twee elementen terug. Voor gehele getallen is het verschil tussen twee opeenvolgende elementen uiteraard het verschil in grootte. Voor C-strings bijvoorbeeld kan dit het verschil in lengte zijn maar dat zou ook de afstand tussen twee woorden in een tekstbestand kunnen zijn. Je geeft dus de nodige functie die de afstand tussen twee elementen kan bepalen als parameter mee. Wanneer de array minder dan twee elementen telt, wordt de nulpointer teruggegeven.

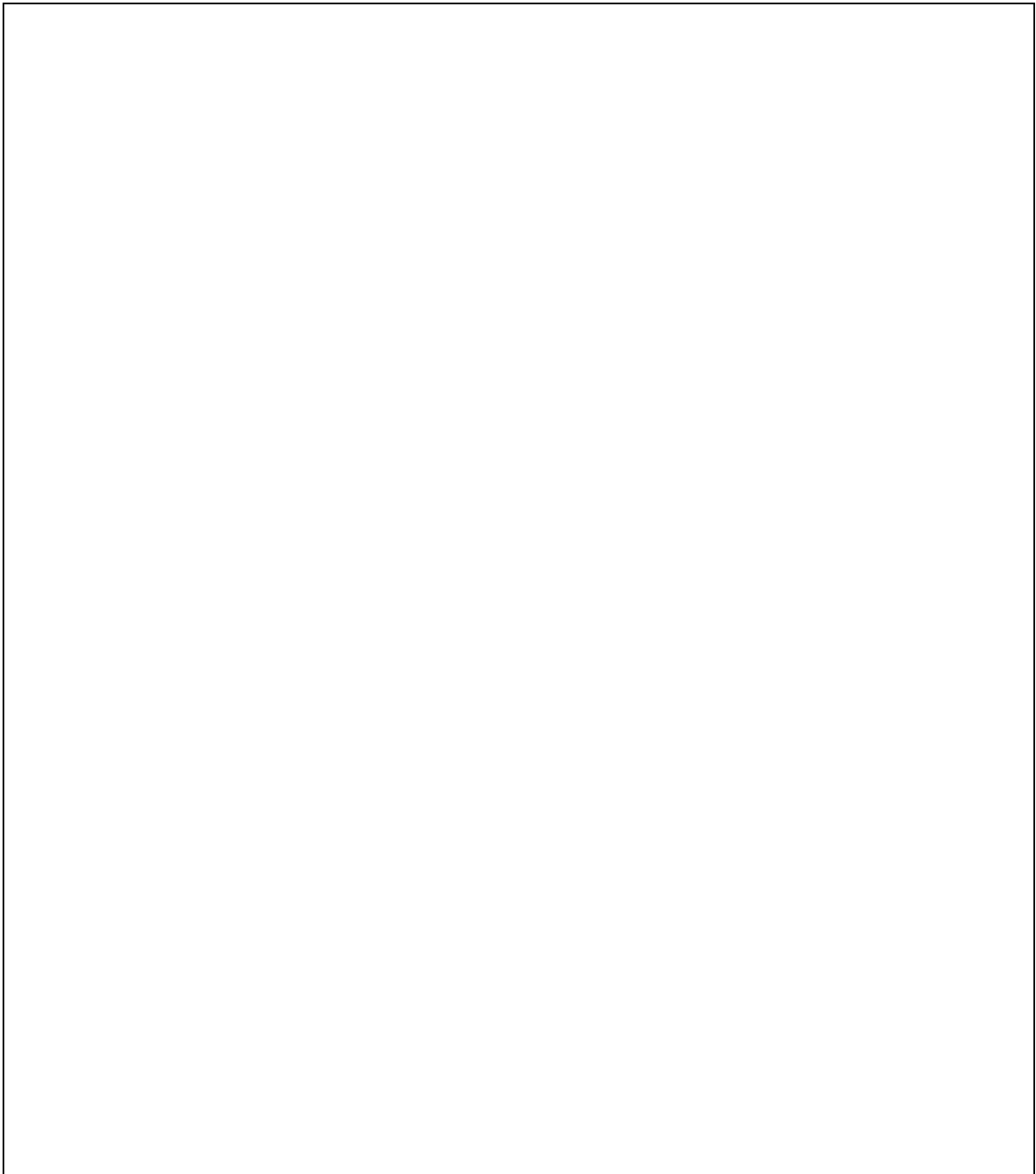
Schrijf de algemene functie **bepaal_grootste_vershil**.

3)

- a) Gegeven in een hoofdprogramma een C-string bestaande uit woorden (bestaande uit enkel kleine letters) die van elkaar gescheiden zijn door één of meerdere spaties. Gegeven ook een pointer ***p*** die wijst naar een karakter van deze C-string. Schrijf een functie ***geef_woord(p)*** die uit deze C-string het eerste woord (beginnende vanaf ***p***) “knipt” en dit woord als resultaat teruggeeft. Wanneer deze C-string vanaf ***p*** geen woord (meer) bevat, wordt de nulpointer teruggegeven. De pointer ***p*** wordt verplaatst naar het karakter net voorbij het teruggegeven woord (of dus het null-karakter als de C-string geen woord meer bevat).

Bijvoorbeeld:

Wanneer ***p*** wijst naar het eerste karakter van de tekst “_een_ _zin”, geeft de functie een nieuwe C-string met als inhoud “een” terug en wordt ***p*** verplaatst naar de volgende spatie.



- b) Schrijf een functie **geef_woorden(s)** die een array maakt (en teruggeeft) met daarin de eerste 10 woorden van **s** (of minder indien **s** geen 10 woorden bevat). Na het laatste woord bevat de array de nulpointer. Voorzie net voldoende plaats voor de array. Gebruik uiteraard voorgaande functie.

Vervolg net (gelieve voorafgaand wel het nummer van de vraag te noteren)