

Structuur van Computerprogrammas II: Week 5

Elisa Gonzalez Boix Maarten Vandercammen Robbe De Greef

1 Enumerations

1. (**Dodona**) Schrijf een enumeration voor de booleans TRUE en FALSE. Implementeer daarna zelf de booleaanse functies
`enum Boolean my_and(enum Boolean a, enum Boolean b)` en
`enum Boolean my_or(enum Boolean a, enum Boolean b)` door gebruik te maken van de operatoren + en *.

2 Pointers en pointer arithmetic

2. (**Dodona**) In C bevat de `<string.h>` library de functie `char *strrchr(char *string, char c)` die een pointer teruggeeft naar het laatste voorkomen van het karakter `c` in `string`. Als dat karakter helemaal niet voorkomt in de string, geeft de functie NULL terug.

Als je de functie bv. op de volgende manier oproept:

```
char *p = strrchr("www.vub.be", '.');
```

wijst `p` naar het tweede punt in de url, waardoor `printf("top-level domeinnaam is %s", p);` het volgende print:
`domeinnaam is .be.`

Implementeer nu je eigen versie van deze functie:

```
char *my_strrchr(char *string, char c)
```

3. (**Dodona**) De functie `char *strpbrk(char *string1, char *string2)` uit de `<string.h>` library controleert of er zich ergens in `string1` een karakter bevindt dat ook voorkomt in `string2`. Als dat zo is, geeft de functie een pointer terug naar het eerste karakter uit `string1` dat ook in `string2` voorkomt. Als `string1` geen karakters bevat die ook in `string2` staan, geeft de functie NULL terug. Als je de functie bv. op de volgende manier oproept:

```
char *p = strpbrk("Voorbeeld", "aeiou");
```

wijst `p` naar de eerste `'o'` uit `"Voorbeeld"`.

Implementeer nu je eigen versie van deze functie:

```
char *my_strpbrk(char *string1, char *string2)
```

4. (**Dodona**) In C kan je de `strcmp` functie uit de `<string.h>` library gebruiken om twee strings met elkaar te vergelijken om na te gaan welke van de twee alfabetisch voor de andere komt. Deze functie geeft een negatief getal terug als de eerste string alfabetisch voor de andere komt, een positief getal in het andere geval, en 0 als beide strings identiek zijn. `strcmp("Scheme", "Scala")` geeft bv. het getal 7 terug om aan te geven dat "Scala" voor "Scheme" in het alfabet komt. Merk op dat er 7 wordt teruggegeven omdat h 7 plaatsen verder in het alfabet staat dan de a. Een ander voorbeeld: `strcmp("hello", "hellow")` geeft -119 terug aangezien de eerste string alfabetisch voor de tweede komt en de ascii waarde van 'w' 119 is.

Schrijf nu je eigen versie van deze functie:

```
int my_strcmp(char string1[], char string2[]).
```

 Om te tellen hoeveel plaatsen er tussen twee karakters staan, kan je deze karakters gewoon van elkaar aftrekken: `'h' - 'a'` geeft bv. 7.

5. Bekijk onderstaande code en beantwoord of deze definities en assignments toegelaten zijn of een compiler of run-time fout produceren.

```
char p[] = "hallo";
char *p2 = p + 1;
*p2 = 'q';

char *p3 = "hallo";
char *p4 = p + 1;
*p4 = 'q';

char const s1[] = "hallo";
s1[2] = 'J';

char const *s2 = "hallo";
*s2 = 'J';

char const *u = "hallo";
char const *v = "world";
u = v;
```

3 Extra

6. (**Dodona**) Implementeer de functie `char *string_span(char *string, char *accepted, int *span_size)` die de grootste *span* van aanvaarde karakters in de parameter `string` zoekt. De span is de grootste substring

in `string` die enkel bestaat uit karakters die voorkomen in `accepted`. `string_span` geeft een pointer terug naar het begin van die span en het slaat de lengte van de span op in de integer waar de pointer `span_size` naar wijst (gelijkaardig aan hoe je in oefening 5 van het vorige WPO de oplossingen van een vierkantsvergelijking hebt opgeslagen in variabelen waar pointer parameters naar wezen). Als er geen span bestaat, geef je een pointer terug naar het einde van `string`.

Voorbeelden:

```
char string1[] = "abcdefg";
char accepted1[] = "fde";
int length1;
char *span1 = string_span(string1, accepted1, &length1);
// length1 = 3 en span1 wijst naar het karakter 'd' in string1

char string2[] = "ab_cdefg_cadg";
char accepted2[] = "abcdef";
int length2;
char *span2 = string_span(string2, accepted2, &length2);
// length2 = 4 en span2 wijst naar het eerste karakter 'c' in string2
```