

C

Examenvragen

Master of Science in de Industriële
Wetenschappen

Industrieel Ingenieur

Examen C (xx/01/2016) (versie 1)

Vraag 1)

Gegeven is volgende array t en pointers p, q, r (deze zijn correct geïnitialiseerd).

Duid aan of de bewerkingen juist of fout zijn. Als deze juist zijn, schrijf dan nauwkeurig wat er gebeurd. Als ze fout zijn, vermeldt wat er mis is.

```
int t = {1, 2, 3, 4, 5}, const int * q, int * const * p, int * r  
*q *= *t  
*r = *t++  
**p = *t  
*p *= t
```

Vraag 2)

Schrijf een procedure eerste_bit(g) die een meegegeven positief geheel getal g verandert: In hexadecimaal notatie moet het meest beduidende cijfer van het getal een F worden. Dus 'A8E' wordt 'F8E'. Gebruik enkel bitoperatoren, dus geen bewerkingsoperatoren (+, *, %...) noch conditionele operatoren (==, >, !=...).

Vraag 3)

Schrijf een functie codeer(s) die een meegegeven c-string s codeert en het gecodeerde c-string teruggeeft. Voor de codering wordt simpelweg voor elk karakter het aantal keren dat het voorkomt daarna het karakter zelf, voorgesteld.

Bv.: AAAdRRfffff -> 3A1d2R5f. Het c-stringresultaat moet net groot genoeg zijn.

Vraag 4)

Implementeer een recursieve functie tel voorwaarde(t, func) die het totaal aantal keren teruggeeft dat een meegegeven functiepointer func true teruggeeft indien je die functie oproept voor elke c-string van een (eveneens) meegegeven array van c-strings t. Hoe doe je dit zonder de size van de array mee te geven?

Vraag 5)

Gegeven een struct knoop met twee elementen: int getal en knoop * next.

Implementeer de procedure die de i-de knoop van een meegegeven gelinkte lijst verwijdert. (dus parameters zijn int i en een gelinkte lijst). Indien i groter is dan het aantal knopen, moet er niets gebeuren.

Examen C (xx/01/2016) (versie 2)

Vraag 1)

Gegeven is volgende array t en pointers p, q, r (deze zijn correct geïnitialiseerd). Duid aan of de bewerkingen juist of fout zijn. Als deze juist zijn, schrijf dan nauwkeurig wat er gebeurd. Als ze fout zijn, vermeldt wat er mis is.

```
int t [];  
int *p; const int *q; int const * r; int*const*s;  
p = *t++;  
...
```

Vraag 2)

Bitoperatoren: schrijf een functie reverse die een octaal getal omdraait. Dus als je 5637 ingeef, krijg je 9365 terug. Hierbij mag geen gebruik gemaakt worden van wiskundige operatoren +; -; /; * maar enkel van bitoperatoren.

Vraag 3)

Gegeven een geordende lijst van C strings en je moest een C string inlezen van max 80 karakters en deze in de lijst toevoegen zodat hij alfabetisch geordend bleef (zie hoofdstuk 4 de theorievoorbeelden).

Examen C (xx/01/2017) (versie 1)

Vraag 1) (4 punten)

Gegeven is volgende array t en pointers p, q, r (deze zijn correct geïnitialiseerd).

Duid aan of de bewerkingen juist of fout zijn. Als deze juist zijn, schrijf dan nauwkeurig wat er gebeurd. Als ze fout zijn, vermeldt wat er mis is.

```
int t[5] = {1,2,3,4,5}; const int *p, int* const *q, int *r
```

```
**q=(*t)++;
```

```
*r *= *p++;
```

```
r=p;
```

```
*q = p +1;
```

Vraag 2) (2 punten)

Bitoperatoren: a=geheel getal, b=variabele int

-Zet 4 laatste bits van a op 0, rest ongewijzigd

-Stop voorlaatste bit van a in b, a ongewijzigd

ANTWOORD: a &= ~15; b = (a & 2);

Vraag 3) (9 punten)

Gegeven: array t met n strings (stijgend, alfabet gerangschikt)

Schrijf: procedure lees_en_voeg_toe(t,n) die 1 woord inleest (max 80 char)

En deze C-string dan toevoegt, t blijft gerangschikt, onderstel t groot genoeg, ook n aanpassen

Vraag 5) (9 punten)

Schrijf: procedure verwijder_letters(s,r), s en r zijn 2 C-strings met kleine letters, alle letters van r zijn verschillend, procedure wijzigt s door alle letters van r te verwijderen uit s (alfabet -> _____ als 1= "bal")

! strings slechts 1 keer overlopen

! verplicht schuivende pointers

! geen string.h functies

Tip: je moet geheugen ruimte van verwijderde letters nt (NIET? Of wel?) vrijgeven

Vraag 6) (6 punten)

Gegeven: Gelinkte lijst declaratie

Schrijf: procedure zet_eerste_knoop_achteraan(l) Verplaatst 1^e knoop van l naar achter, indien minder dan 2 knopen: niet gebeurd

<https://www.geeksforgeeks.org/move-first-element-to-end-of-a-given-linked-list/>

Examen C (xx/01/2017) (2^{de} zit)

C/C++ , september 2017

KLAD

① C (9 ptn)

gegeven: array t met n strings (stijgend, alfabet. gerangschikt)

schrijf: procedure lees_en_voeg_toe(t, n), die 1 woord inleest (max 80 char)
en deze C-string aan toevoegt, + blijft gerangschikt!
onderstel + groot gescrewd, ! ook & oanpassen

② C (6 ptn)

gegever: gefilterte lijst declaratie

schrijf: procedure zet_eerste_door_achteraan(l)

→ verplaatst 1e kroog van l naar achter, indien minder dan 2 kroogen: niets gebeurt.

③ C (4 ptn)

gegeven: int t[5] = {1, 2, 3, 4, 5}; const int *p; int *const q; int *n;

Waar/Font + uitbreng: $*n = *q = (*t) ++$

$$*n = *p = *p ++;$$

$$*n = p;$$

$$*q = p + 1;$$

④ C (9 ptn)

schrijf: procedure vervangen - letters (r, s), een r en 2 C-strings met kleine letters
alle letters van r zijn verschillend, procedure wijzigt r door alle letters
van r te vervangen uit s (alfabet \rightarrow het als 1 = "bol")

| strings slechts 1 keer overlopen

| inhoudende pointers voldoet

| geen string-h functies

tip: je moet gebruik maken van verwijzend letters niet vrijgeven

⑤ C (2 ptn) bits, enkel bitoperatoren en =

a = geheel getal, b = variabele int

- zet h laaste bits v a op 0, rest ongevoerd

- stop voorlaatste bit v a in b, a \ll

Examen C (15/01/2020) (versie 1)

Vraag 1)

Typische vraag over pointers

Vraag 2)

Bitoperatoren: In deze oefening schreef je een programma die controleerde of dat er in een gegeven getal in octale voorstelling een getal meermaals na elkaar voorkwam. Deze gaf het resultaat dan terug als een boolean. Zo is 656 bijvoorbeeld false maar 4553 true. De functie moet verplicht recursief zijn.

Vraag 3)

Je moet een programma schrijven die van een gegeven string een gecodeerde string teruggeeft maar dan in Run Length Encoding. De teruggegeven string moet juist groot genoeg zijn.

Bv: AAAA//nHH wordt dan 4A2/1n2H

Vraag 4)

Kreeg een struct van een complex getal:

```
typedef struct {  
    int re, im;  
} complex;
```

En moest hiermee met Horner de veelterm evalueren in x. Het resultaat moet dan weer een complex getal zijn

Vraag 5)

Er werden twee verschillende structs gegeven, een van een rechthoek met een hoogte en breedte en een van een gekleurde rechthoek. Je moest dan een programma schrijven die in een collectie van rechthoeken een pointer verschuiven die oorspronkelijk op de eerste stond verschuiven naar een rechthoek met de gegeven data. Als deze er niet was dan deed je niets. (moest het dan niet de nullptr worden?)

Vraag 6)

Gegeven: Gelinkte lijst declaratie

Schrijf: procedure zet_eerste_knoop_achteraan(l) Verplaatst 1^e knoop van l naar achter, indien minder dan 2 knopen: niet gebeurd

Examen C (15/01/2020) (versie 2)

Vraag 1) (4 punten)

Typische vraag over pointers

Vraag 2) (5.5 punten)

Bit manipulation: Schrijf een functie verwijder_octaal_cijfer(int g, int c) met g een integer en c een enkel octaal cijfer (dus $0 < c < 8$). De functie verwijdert uit de octale voorstelling van g alle octale getallen die gelijk zijn aan c. Dus als de int g in octale voorstelling 13323 is en c is 3, dan moet het resultaat van de functie 12 worden. De functie moet verplicht recursief geschreven worden. Maak GEEN veronderstellingen over het aantal octale cijfers van g, gebruik enkel bit operatoren.

Vraag 3) (2.5 punten)

Gegeven:

Gevraagd: geef de inhoud van de array t.

```
int t[5] = {2};  
int *pt[2];  
int **q = &pt[0];  
*pt = t + *t;  
pt[1] = *pt + 1;  
**q = -4;  
**(pt+1) = 6;
```

Vraag 4) (10 punten)

Schrijf een functie verwijder_uit_string(const char *s, const char *r) die alle letters uit s verwijdert die ook in string r voorkomen. Beide strings bestaan enkel uit kleine letters. Beide strings mag je hoogstens 1x overlopen. Het resultaat is een c-string die juist groot genoeg is.

Vraag 5) (9punten)

```
Gegeven is  
struct persoon{  
    int leeftijd;  
    char naam[50];  
}persoon;
```

Gegeven is een array t van personen. De array is gerangschikt op naam en daarna op leeftijd. De array bevat n personen.

schrijf een procedure lees_en_voeg_toe(t, n) die een nieuwe persoon toevoegt aan de array t. Lees eerst een naam in en vervolgens een leeftijd (beide inputs moeten niet gecontroleerd worden). Voeg de nieuwe persoon toe aan de array en zorg hierbij dat de array gerangschikt blijft. Eerst wordt alfabetisch op naam gesorteerd en als de naam gelijk is wordt op leeftijd gesorteerd (kleinste eerst). Pas ook het aantal n aan! (n moet dus pointer zijn). Ga er van uit dat de array t zeker groot genoeg is.

Vraag 6)

Schrijf een procedure verzet_laatste_knoop(l, f) die van de laatste knoop de eerste maakt als het getal van deze laatste knoop voldoet aan de logische functie f. Als de lijst uit minder dan 2 knopen bestaat hoeft uiteraard niks te gebeuren.

Examen C (.../01/2021)

Vraag 1)

Typische vraag over pointers

Vraag 2)

Schrijf een recursieve functie max_rec(t,n,...) die het grootste element bepaalt van de meegegeven array t. Het datatype van de array is niet gekend. De functie geeft een pointer terug naar het grootste element. Het grootste element wordt bepaalt a.d.h.v. een meegegeven functie.

Vraag 3)

Schrijf een procedure 'wissel_knopen(p,i) die de ide knoop van de lijst wisselt met de (i+1)de knoop.

Vraag 4)

Geg:

```
Typedef struct{
    double lengte;
    double breedte;
}Rechthoek;
```

```
Typedef struct{
    Rechthoek r;
    char kleur[50];
}GekleurdeRechthoek;
```

Gevr: Schrijf een procedure verplaats_ptr(gr, n) die aan de gebruiker een kleur vraagt. De ingegeven kleur wordt gecontroleerd of deze aanwezig is en mag maximaal 50 karakters bevatten. De gevonden gekleurde rechthoek moet naast de gevonden kleur, eenzelfde lengte en breedte hebben.

Vraag 5)

Schrijf een logische functie lees_in_controleer(g) die aan de gebruiker vraagt om een octaal getal in te geven en deze vervolgens controleert of het ingegeven getal een octaal getal is. Het ingegeven octaal getal moet achtereenvolgend dezelfde cijfers bevatten.

Vraag 6)

Schrijf een functie encode(t,k) die een meegegeven C-string t “ontcijfert” door gebruik te maken van een Caesar sleutel k. Hierbij worden de karakters gewisseld volgens de sleutel k. Bv. k=2, cs= “Hallo!”, dan zal de C-string het volgende worden “jcnnq!”. Hoofdletters worden omgezet in kleine letters en leestekens blijven onveranderd. De nieuwe gevormde c-string wordt als resultaat teruggeven.

C++

Examenvragen

Master of Science in de Industriële
Wetenschappen

Industrieel Ingenieur



Examen C++ (xx/01/2016) (versie 1)

Vraag 1)

Schrijf een procedure verwijder(m, v) met map<T, set<S>> m en vector<S> v die de elementen uit v verwijdert die niet in alle sets aanwezig zijn.

*(je moet de vector overlopen en dan kijken of de map de sleutel bevatte, daarna kijken of de sleutel een bijhorende waarde had. Zo niet-> wissen.) *fantastische bijdrage**

Vraag 5)

Schrijf een functie die door middel van 2 parameters (een string die de bestandsnaam voorstelt en een logische functie) die de eerste regel teruggeeft waarvoor de meegegeven functie waar/geldig blijkt te zijn. Schrijf eveneens een main-programma waarbij je deze functie gebruikt met een bestand “file.txt” en een lambda functie die test of het woord gelijk is aan het woord “verjaardag” (ben niet zeker of het “verjaardag” was, maar het maakt niet veel uit). Zorg dat die stringresultaat uitgeschreven wordt. Hierbij moet er een exceptie worden geworpen als het bestand niet kon worden geopend en als het woord niet aanwezig is. In de main moet deze exceptie dan worden opgevangen (staat niet vermeld in de opgave).

Examen C++ (xx/01/2016) (versie 2)

Vraag 1)

- a. operatoren van die vector overschrijven, niet implementeren:
- i) constructor
 - ii) push_back
 - iii) operator +=
 - iv) copy-constructor
 - v) operator <<
 - vi) operator >>
 - vii) ?
- b. definitie van een klasse met vector, gegeven main, schrijf push_back()

Vraag 2)

Schrijf een functie keerom(m,om) waar je in een map de sleutels van een specifieker type T afbeeldt op een type S omdraait zodat de elementen van S de sleutels zijn de elementen van T de values.

Vraag 3)

Geef declaratie van vergelijk functie die werkt met een functiepointer. De functies waar je naar wijst waren *is_even* en '?' die als parameter een integer meekrijgen en vertellen of het getal even is of niet.

Vraag 4)

A & B afgeleide klasse van A. Hebben allebei schrijf() methode en je moet void proc(?) met 1 parameter schrijven. Implementeer ook het minimale dat je nodig hebt in de class A en B.

Vraag 5)

geef de output van een recursieve functie (een soortgelijke staat in powerpoint slide 47)

```
void proc(int a, int b) {  
    if(a < b)  
    {  
        printf("%d %d", a,b);  
    }  
    else  
    {  
        proc(a, b += 2);  
        printf("a = %d b = %d",a,b);  
        proc(a--,b++);  
    }  
}
```

Vraag 6)

Je moet een bestand inlezen en de laatste tien regels (of als minder als het bestand minder regels bevat) in een uitvoerbestand steken.

Examen C++ (xx/01/2017) (versie 1)

Vraag 1)

/

Vraag 2.1) (a)

Schrijf een functie geef_getallen (bn , n, fn) die via het hoofdprogramma een tekstbestand "verhaal.txt" meekrijgt en hieruit de laatste n getallen in omgekeerde volgorde van inlezen teruggeeft. De functie controleert bovendien ook of deze laatste n getallen voldoen aan de meegegeven functie fn. Indien het bestand minder dan n getallen bevat wordt er een logic error opgeworpen "te weinig getallen".

Vraag 2) (b)

Schrijf het hoofdprogramma voor de vorige functie en maak hierbij gebruik van lambda functies. Het hoofdprogramma krijgt 2 files (s,d) mee via de command line.
De voorwaarde van de functie zijn als volgt:

Voeg laatste 10 getallen van s die strikt positief zijn
en deelbaar door 3 OF 5 toe aan bestand d (met vorige functie)

Vang als laatste bijhorende logic errors op als s niet geopend kan worden

Vraag 3)

Geg: Klasse B is afgeleid van klasse A

```
int main() {  
    A a;  
    A a2(5);
```

```
    B b;  
    B b2(1);  
    B b3(2, 3);
```

```
    return 0;
```

}Gevraagd: Schrijf de klasse declaratie van A en B en bijhorende definities.

Vraag 4)

Schrijf een procedure keer_om(mp, omp) waar je in een map de sleutels van een specieker type T afbeeldt op een type S omdraait zodat de elementen van S de sleutels zijn de elementen van T de values.

Vraag 5)

Geg:

Klasse mijn_array:

```
int main(){
    mijn_array a1;
    cin>>a1;
    a1[3]++;
    a1=a1*2;
    return 0;
}
```

Gevraagd: public deel van Array zodat dit werkt

Examen C++ (xx/01/2017) (2^{de} zit)

VERVOLG ANTWOORD OP VRAAG 1

⑥ C++ (6 ptn)

gegeven S1 May m beeldt typeT af op element van typeS

schrijf void lees_om(m, m) die 'm' moet: omgekeerde S1 May: beeldt elementen v types af op sets van type T

⑦ C++ (5 ptn)

gegeven: afleidtijds kleine Array en main:

→ schrijf public deel v array zodat dit werkt

Array v1;

Cin > v1;

v1[3] ++;

v1 = v1 * 2;

return 0;

⑧ C++ (6 ptn)

schrijf klasse A en B zo compact mogelijk

→ A: 1 private int : a

default const. (a → 3)

const. met 1 param → a

fnct void schrijf () : schrijft "a = ..."

B: afgeleid v A

attribut int b

default const. (a → 3, b → 5)

const. 1 param → a, b → 5

 (1 2 param → a, → b)

overrupt fnct void schrijf

→ uitwendige procedure: kiezen

void printab(?) 1 param

schrijft op 1 param de procedure
van A of B uit

(9) C++ (7 ptn)

gegeven: bestand s met enkel # gebele getallen (kan grote rinden hebben)

schrijf: geef -getallen (s, n, func) : func: logisch file niet param int

↳ geeft collectieve terug met de laatste n getallen uit s die volgen aan func

in reverse volgorde, en kunnen ook minder dan n getallen in s zitten

+ exceptie voor bestand s

⑩ C++ (6 ptn): schrijf main die via command 2 files meekrijgt: s, d

→ voegt laatste 10 getallen van s die strijd passen op delbestand 3OF5 toe aan bestand d (metvorige functie)

verplicht met lambda functie
laat als middel op meer strings op command line staan en als s niet gevonden

Test C++ (/2018)

Vraag 1)

/

Vraag 2)

/

Vraag 3)

Geg:

Klasse Verzameling:

```
int main(){
    Verzameling verz1, verz2;
    cin >> verz1
    verz1 -= 'c';
    verz2 = 'a' + verz2;
    Verzameling doorsnede = verz1 * verz2;
    verz2 = verz1++;
    cout << - verz2
    map<Verzameling, Verzameling> relatie;
    relatie[verz1] = verz2;
return 0;
}
```

Gevraagd: Schrijf de '-' operator en '++' operator van de klasse Verzameling.

Vraag 4)

Geg:

```
class ?
private:      set<vector<char*>*>s;
               int tabelgrootte;
public:
    zelf...
```

Gevraagd: toekenningsoperator en deconstructor, gegeven dat char * niet noodzakelijk een c-string is maar een chartabel met grootte tabelgrootte;

Vraag 5) (a)

Gegeven is het hoofdprogramma van de klasse Bibliotheek

Opm: boeken bevat een map die genres bevat, die map bevat met auteurs, die een multiset bevat met titels. Let op b1-=b1 moet niks doen!

```
int main(){
    Bibliotheek b1,b2;
    ???
    b1-= b2;
    b1-=b2;
    return 0;
}

class Bibliotheek {
    private:
        map<string,map<string,multiset<string>>> boeken;
    public:
        zelf...
```

Gevraagd: schrijf de klasse Bibliotheek zodat de voorgaande opdrachten werken, je moet ze niet te implementeren!

Vraag 5) (b)

Schrijf voor

???

de bijhorende code voor de -= operator.

Examen C++ (15/01/2020) (versie 2)

Vraag 1)

Schrijf de klasse LangeSlang (zie Reeks E oef 26)

Vraag 2) (a)

Schrijf een logische functie controleer_woord(bn , n, fn) die via het hoofdprogramma een tekstbestand “verhaal.txt” meekrijgt en hieruit de laatste n woorden controleert of ze voldoen aan de meegegeven functie fn.

Vraag 2) (b)

Schrijf het hoofdprogramma voor de vorige functie en maak hierbij gebruik van lambda functies. De voorwaarde van de functie zijn als volgt:

- Woorden moeten met een kleine letter starten,
- Woorden moeten eindigen met een vraagteken en
- mogen “ex” niet bevatten als substring.

Vraag 3)

/

Vraag 4)

Geg: map<int, map< string, set<string>> personen;

Gegeven is een map (personen), die een meegegeven geboortejaar (int) afbeeldt op een map van familienamen (string) die de familienamen op hun beurt afbeelden op een set (strings) die de voornamen van de geborenen van dat jaar en dus van die familienaam bevatten.

Schrijf de functie geef_voornamen(personen, familienaam) die de voornaam van de persoon uit wiens familienaam “Jansens” is en die in minstens 5 verschillende jaartallen voorkomt. Als er geen voornaam minstens 5 keer voorkomt, wordt een lege string teruggegeven.

Vraag 5)

a)Files

Je kreeg een struct van een file, deze houdt een bestandsnaam bij.
Deze kon je uitschrijven (de inhoud ervan) met behulp van <<

b)Files met auteur naam

Hier schreef je autfile afgeleid van file, deze hield naast een bestandsnaam ook een gebruikersnaam bij. Deze kon je uitschrijven als de bestandsnaam niet begon met een ., als deze wel begon met een punt dan moest je het bestand uitschrijven zoals bij een gewone file. (functie van ouder gebruiken dus)

c)Meerdere files stack

je moest meerdere files van beide types kunnen opslaan (op de)/(in de) stack, zelf voegde je hier ook nog een file naar keuze aan toe. In de veronderstelling dat er meerdere files in de stack zaten, moest je de stack ook printen (dynamic binding moet dus mogelijk zijn).

-> oplossing: via unique_pointers. Bij het printen kopie maken van de stack!

-> oplossing a en b: overschrijf 1x << in basisklasse (niet als friend), en schrijf in de beide klassen een virtuele, publieke functie print. << operator maakt dan gebruik van print.

Vraag 6)

a)Vectoren

Geef de declaratie van bepaalde operatoren zoals vb de ++(prefix), ostream, istream, pushback, move operator, ... horende bij een gegeven mainprogramma (zelf afleiden welke je moet schrijven).

b)Pushback, move

schrijf de implementatie van deze functie uit. De initialisatie van deze functies gebeurt in de klasse van bij a).

Examen C/ C++ (xx/01/2020) (versie 3)

C

- 1 8ptn juist/fout
- 3 4,5ptn zoeken in array van onbekend type en met functieparam.
→ bol, punt
- 3 5ptn bt: recursief, tel hoeveel keer cijfer C in hex getal g
↳ knoop is elke letter verschilles v C-strijg (freq. table)
- 3 4ptn gelijktijdig met punt (struct met double x, y) nieuwe knoop voor
knoop waar x == y.
- 3 7ptn procedure, array v C-strijgs, pointer naar array verstu-
naar C-strijg next offset. Ma niet uitgeleren woord.
-
- 18

C++

- 4 8ptn lange slangs: += operator bongen zig a: ... ^b ^a
- 4 6,5ptn ~~map<string, map<int, std::string>>~~ volleden aan grepmaam,
geboortejaar collect aan logische functie als param
& throw, tel max #namen (grootste & t.)
- 3 3,5ptn verplicht lambda f., exceptions opvangen
- 3 6ptn Klasse Afgeleid v Klasse basis = =, <<, const.
- 4 7ptn Rule of 5, met punten: 5 declarations + toek op def.
-

Examen C++ (.../01/2021)

Vraag 1)

Schrijf de klasse LangeSlang (zie Reeks E oef 26)

Vraag 2) (a)

Schrijf de functie alfabetisch_kleinste(bnm, n, function) die via het hoofdprogramma een tekstbestand "verhaal.txt" meekrijgt en hieruit het alfabetisch kleinste woord bepaalt en vervolgens ook teruggeeft. De functie controleert of het woord n-keer voorkomt en voldoet aan de meegegeven functie

Indien het bestand geen woorden bevat wordt er een logic error opgeworpen "geen woord gevonden".

Vraag 2) (b)

Schrijf het hoofdprogramma voor de vorige functie, waarbij het alfabetisch kleinste woord 5 keer mag voorkomen in het tekstbestand "verhaal.txt". Maak gebruik van lambdafuncties. De voorwaarde van de functie zijn als volgt:

Het woord mag de letter p niet bevatten.

De eerste en laatste letter van het woord dienen hetzelfde te zijn.

Vang als laatste bijhorende logic errors op.

Vraag 3)

Geg:

Klasse Basis:

```
int main(){
    Basis a;      // getal=1
    Basis b(3);   // getal =3
    Basis c(b);
    if(b==c){
        cout<< b << "==" <<c;
    }
return 0;
}
```

Klasse Afgeleid

```
int main(){
    Afgeleid a;          // getal= -2, extra=4
    Afgeleid b(9) ;       // getal = 9, extra =4
    Afgeleid c(9,7);     // getal = 9, extra = 7
    Afgeleid d(c)
    if(b==c){
        cout<< "af" << b << "==" <<c;
    }
return 0;
}
```

Gevraagd: Schrijf de '==' operator en '<<' operator van de klasse afgeleid.

Vraag 4)

Geg. Map<string, map<string, set<int>>> straten;

Gegeven is een map ‘straten’ die een meegegeven straat (string) afbeeldt op een map van kranten(string) die op zijn beurt afgebeeld worden op een set<int> van huisnummers.

Schrijf de functie tel_kranten(straten, krant) die het aantal adressen telt die een gegeven krant moet ontvangen.

Vraag 5) (a)

Gegeven is het hoofdprogramma van de klasse mijn_Vector

```
int main(){
    mijn_vector v1,v2,v3;
    cin>>v1;
    v2= move(v1);
    v2++;
    cout <<v2 [3];
    v3=v2-v1;
    return 0;
}
```

```
class mijn_vector{
    private:
        int *t;
        int g;
        int n;
    public:
        zelf...
```

Gevraagd: schrijf de klasse mijn_Vector zodat de voorgaande opdrachten werken, je moet ze niet te implementeren!

Vraag 5) (b)

Schrijf voor

```
v2= move(v1);
v2++;
```

de bijhorende code voor de move constructor en ++ operator.

C

Examenvragen opl

Master of Science in de Industriële
Wetenschappen

Industrieel Ingenieur



Examen C (xx/01/2016) (versie 1)

Vraag 2)

```
int main(){
    int g = 0xa8e;
    printf("%0x ", g);
    eerste_bit(&g);
    printf("%0x ", g);
    return 0;
}

/*
-----*/
void eerste_bit(int *g) {
    int a = 0xF;
    int b = *g>>4;      /* Verwijderen van de Laaste 4 bits van '*g' */

    while(b!=0){          /* Hoe groot is het getal "g" ? */
        b = b>>4;        /* Verwijderen van de Laaste 4 bits */
        a = a<<4;         /* Plaastmaken voor 4 bits */
    }
    *g = (*g|a);          /* Toevoegen van het meest signifante cijfer aan '*g' */
}
```

Vraag 3)

```
int main(){
    char cs[81] = "AAAAA/nHH";
    const char *pcs;
    pcs= encode(cs);
    puts(cs);
    puts(pcs);

    return 0;
}

/* -----
char *encode(const char *pcs_c){

    char cs[MAX_A_C+1];
    int sizeMi;
    int sizeDT;
    int sizeH;
    char *pcs;
    int aantal;

    int i;
    int j = 0;
    int k;

    sizeMi = strlen(pcs_c);
    sizeDT = sizeof(char);
    sizeH = (sizeMi+1)*sizeDT;
    pcs = (char*) malloc(sizeH);

    /* Traverse the input string one by one */
    for (i=0; i<sizeMi; i++) {

        /* Copy the first occurrence of the new character */
        pcs[j] = pcs_c[i];
        j++;
        aantal=1;
        /* cs the number of occurrences of the new character */
        while ( i+1<sizeMi && pcs_c[i] == pcs_c[i + 1]) {
            aantal++;
            i++;
        }

        /* Store aantal in a character array cs[] */
        sprintf(cs, "%d",aantal);

        /* Copy the cs[] to destination */
        for (k = 0; *(cs + k)!=0; k++) {
            pcs[j] = cs[k];
            j++;
        }
    }

    /*terminate the destination string */
    pcs[j] = '\0';
    return pcs;
}
```

Vraag 5)

```
/* ----- */
void deleteIdeKnoop(knoop **ppk, int n){
    knoop *hpk1 = (*ppk);
    knoop *hpk2;
    knoop *hpk3;
    int i=1;

    while(hpk1->next!=0 && i<n){
        hpk2 = hpk1;
        hpk1 = hpk1->next;
        i++;
    }
    hpk3 = hpk1->next;
    printf("Ik geef knoop %d vrij\n",hpk1->getal); /*ter info*/
    free(hpk1);
    hpk2->next = hpk3;
}
```

Examen C (xx/01/2016) (versie 2)

Vraag 2)

```
int main(){
    int a,b;
    a = 05637;
    printf("Original: \n");
    printf("a is = %o\n",a);

    printf("Reversed: \n");
    b= reverseOctaal(a);
    printf("a is = %o",b);

    return 0;
}

/* -----
int reverseOctaal(int g) {
    int res = 0;
    int hulp =0;

    while(g!=0){
        hulp = g&07; /* Opslaan van het minst signifante cijfer */
        res = res<<3;
        res = res|hulp; /* Toevoegen van het minst signifante cijfer aan 'res'*/
        g = g >> 3; /* Verwijderen van het minst signifante cijfer uit het originele getal 'g'
    }
    return res;
}
```

Examen C (xx/01/2017) (versie 1)

Vraag 2)

```
/* ----- */
uint zet_laatste4bits_opnul(uint x){
    /* x = OUD GETAL */
    /* n = NIEUW GETAL */
    /* h = HULP GETAL */

    uint n;
    uint h1;
    uint h2;

    h1 = 0xF;          /* .... .... .... 1111 */
    h2 = ~h1;         /* 1111 1111 1111 0000 */
    n = x&h2;

    return n;
}
```

Vraag 3)

```
int main() {
    char* acs[5] = {"appel","bannaan","peer"};
    int n;

    n = sizeof(acs)/sizeof(char*);
    printArray((const char *const*)acs,n);

    lees_en_voeg_toe((const char **)acs,&n);

    n = sizeof(acs)/sizeof(char*);
    printArray((const char *const*)acs,n);

    return 0;
}

/*
void printArray(const char* const* c_papcs_c, int n){
    int i;
    for(i=0;i<n;i++){
        printf("%s, ",*c_papcs_c);
        c_papcs_c++;
    }
}
*/
void lees_en_voeg_toe(const char **papc_c, int *pn){

    /* Data storage */
    char cs[MAX_A_C + 1]; /* (1) C-string          */
    int sizeMa;             /* (2) MAX size of C-string */ /* incl. null char */
    int sizeMi;             /* (3) MIN size of C-string */ /* excl. null char */
    int sizeDT;             /* (4) Size of data type   */
    int sizeH;              /* (5) Size of heap        */
    char *pcs;               /* (6) CharPointer voor geheugen op te slaan op de heap */

    int bi;                  /* index van eerste waarde */
    int i;                   /* index van invoegen      */
    int ei;                  /* index van de laatste waarde */
    int x;

    /*
    /* (1) Input */
    printf("\nGeef een tekst: ");

    /* (2) MAX size of C-string */
    sizeMa = MAX_A_C + 1;
    fgets(cs,sizeMa,stdin);

    /* (3) MIN size of C-string */
    sizeMi = strlen(cs);

    /*
    /* (*) Optioneel
    /*Controleer of het voorlaatste char (opm. laatste is null char) een enter is */
    if(cs[sizeMi-1]== '\n'){
        cs[sizeMi-1] = 0;
        sizeMi--;
    }
    /* De buffer dient leeg gemaakt te worden indien... */
    else{
        fflush(stdin);
    }
}
```

```

/* ----- */

/* (4) Size of data type      */
sizeDT = sizeof(char);

/* (5) Size of heap          */
sizeH = (sizeMi +1)*sizeDT;

/* (6) Geheugen reserveren op de heap => pcs voor nodig */
pcs = (char*)malloc(sizeH);

/* (7) Geheugen dat gereserveerd is op de heap opvullen */
strcpy(pcs,cs);

/* ----- */
/* Array overlopen van L-> R: lineair zoeken naar index i (alfabetisch geran.)*/
bi = 0;
ei = *pn;
x = strcmp(papc_c[bi], pcs);

while (bi<ei && x<0) {
    bi++;
    x = strcmp(papc_c[bi], pcs);
}
i=bi; /* de index is gevonden */

/* ----- */
/* Array overlopen van R-> L: lineair zoeken naar index i en de rest opschuiven nr R */
while (i < ei) {
    papc_c[ei] = papc_c[ei - 1];
    ei--;
}
papc_c[i] = pcs; /* de waarde (char) mag toegevoegd worden */

/* ----- */
/*size aanpassen*/
*pn += 1;

}

```

Vraag 4)

```
int main(){
    char cs_d[] = "pbrofdicieagth";
    char cs_s[] = "bdeghj";

    puts(cs_d);
    verwijder_letters(cs_d,cs_s);
    printf("\n");
    puts(cs_d);

    return 0;
}

/* -----
void verwijder_letters(char *pacs_d, char *pacs_s){
    int an[26]={0};
    char *hpacs= pacs_d;

    while(*pacs_s!=0){
        an[*pacs_s-'a']=1;
        pacs_s++;
    }

    while(*hpacs!=0){
        if(an[*hpacs-'a']==0){
            *pacs_d= *hpacs;
            pacs_d++;
        }
        hpacs++;
    }
    *pacs_d=0;
}
```

Vraag 5)

```
/* ----- */
void deleteIdeKnoop(knoop **ppk, int n){
    knoop *hpk1 = (*ppk);
    knoop *hpk2;
    knoop *hpk3;
    int i=1;

    while(hpk1->next!=0 && i<n){
        hpk2 = hpk1;
        hpk1 = hpk1->next;
        i++;
    }
    hpk3 = hpk1->next;
    printf("Ik geef knoop %d vrij\n",hpk1->getal); /*ter info*/
    free(hpk1);
    hpk2->next = hpk3;
}
```

Examen C (15/01/2020) (versie 1)

Vraag 2)

```
int main() {
    if(controleerDubbelOctaal(067653)){
        printf("Wel dubbels");
    }
    else{
        printf("Geen dubbels");
    }
    return 0;
}

/* -----
int controleerDubbelOctaal(int n){
    int b1=0;
    int b2=0;

    if((n>>3)==0) {
        return 0;
    }
    b1 = n&07;           /* Opslaan van het minst signifante cijfer */
    b2 = (n>>3)&07;     /* Opslaan van het volgende minst signifante cijfer */
    if((b1^b2)== 0) {    /* Als b1 en b2 gelijk zijn => b1^b2 == 0 */
        return 1;
    }
    controleerDubbelOctaal(n>>3);
}
```

Vraag 3)

```
int main(){
    char cs[81] = "AAAAA/nHH";
    const char *pcs;
    pcs= encode(cs);
    puts(cs);
    puts(pcs);

    return 0;
}

/* -----
char *encode(const char *pcs_c){

    char cs[MAX_A_C+1];
    int sizeMi;
    int sizeDT;
    int sizeH;
    char *pcs;
    int aantal;

    int i;
    int j = 0;
    int k;

    sizeMi = strlen(pcs_c);
    sizeDT = sizeof(char);
    sizeH = (sizeMi+1)*sizeDT;
    pcs = (char*) malloc(sizeH);

    /* Traverse the input string one by one */
    for (i=0; i<sizeMi; i++) {

        /* Copy the first occurrence of the new character */
        pcs[j] = pcs_c[i];
        j++;
        aantal=1;
        /* cs the number of occurrences of the new character */
        while ( i+1<sizeMi && pcs_c[i] == pcs_c[i + 1]) {
            aantal++;
            i++;
        }

        /* Store aantal in a character array cs[] */
        sprintf(cs, "%d",aantal);

        /* Copy the cs[] to destination */
        for (k = 0; *(cs + k)!=0; k++) {
            pcs[j] = cs[k];
            j++;
        }
    }

    /*terminate the destination string */
    pcs[j] = '\0';
    return pcs;
}
```

Vraag 4)

```
typedef struct{
    int re;
    int im;

}complex;

complex horner(const complex *, int, int);

int main(){
    complex astr[]={{{1,2},{3,1}}};
    int n;
    complex c;

    n = sizeof(astr)/sizeof(complex);
    c = horner(astr,n,10);

    printf("Re = %d  Im = %d", c.re, c.im);

    return 0;
}

/* -----
complex horner(const complex *past, int n, int x){
    complex c;
    int i;

    c.re = past[n].re;
    c.im = past[n].im;

    for( i=n-1;i>=0;i--){
        c.re = (c.re)*x + (past[i].re);
        c.im = (c.im)*x + (past[i].im);
    }
    return c;
}
```

Vraag 5)

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <stdbool.h>
#include <math.h>
#define SIZE_ARRAY 4
#define EPSILON 0.000001

typedef struct{
    double lengte;
    double breedte;
}rechthoek;

typedef struct{
    rechthoek r;
    char kleur[50];
}gekleurdeRechthoek;

void verplaats_ptr(const gekleurdeRechthoek**, int);

/*
int main() {
    int g = 4;
    const gekleurdeRechthoek *grC;
    gekleurdeRechthoek *gr;
    gr = (gekleurdeRechthoek*)malloc(SIZE_ARRAY * (sizeof(gekleurdeRechthoek)));

    char *kl1 = "rood";
    char *kl2 = "oranje";
    char *kl3 = "geel";
    char *kl4 = "groen";

    strncpy(gr[0].kleur,kl1,50);
    gr[0].r.lengte = 1.0;
    gr[0].r.breedte= 2.0;
    printf("gekleurdeRechthoek1 \nLengte= %.1f \nBreedte= %.1f \nKleur= %s\n\n", gr[0].r.lengte,
    | gr[0].r.breedte, gr[0].kleur);

    strncpy(gr[1].kleur,kl2,50);
    gr[1].r.lengte = 1.0;
    gr[1].r.breedte= 2.0;
    printf("gekleurdeRechthoek2 \nLengte= %.1f \nBreedte= %.1f \nKleur= %s\n\n", gr[1].r.lengte,
    | gr[1].r.breedte, gr[1].kleur);

    strncpy(gr[2].kleur,kl3,50);
    gr[2].r.lengte = 2.0;
    gr[2].r.breedte= 2.0;
    printf("gekleurdeRechthoek3 \nLengte= %.1f \nBreedte= %.1f \nKleur= %s\n\n", gr[2].r.lengte,
    | gr[2].r.breedte, gr[2].kleur);
}
```

```

strncpy(gr[3].kleur,k14,50);
gr[3].r.lengte = 1.0;
gr[3].r.breedte= 5.0;
printf("gekleurdeRechthoek4 \nLengte= %.1f \nBreedte= %.1f \nKleur= %s\n\n", gr[3].r.lengte,
gr[3].r.breedte, gr[3].kleur);

grC=gr;
printf("\n\ngekleurdeRechthoek \nLengte= %.1f \nBreedte= %.1f \nKleur= %s\n\n", gr[0].r.lengte,
| gr[0].r.breedte, gr[0].kleur);
verplaats_ptr(&grC,g);
printf("\n\ngekleurdeRechthoek \nLengte= %.1f \nBreedte= %.1f \nKleur= %s\n\n", grC[0].r.lengte,
| grC[0].r.breedte, grC[0].kleur);

free(gr);
return 0;
}

/*
void verplaats_ptr(const gekleurdeRechthoek **past, int n) {
int i=0;
char cs[51];
int scf;
printf("Geef een kleur in: ");
scf = scanf("%50s",cs);

while(scf==0){
    fflush(stdin);

    printf("Geef een kleur in: ");
    scf = scanf("%50s",cs);
}
while(i<n && strcmp((*past)->kleur,cs)!=0 && fabs((*past)->r.lengte - (*past)->r.breedte)>EPSILON ){
    printf("%s \n", (*past)->kleur);
    (*past)++;
    i++;
}
}

```

Vraag 6)

```
/* ----- */
void zet_eerste_knoop_achteraan(knoop **ppk) {

    knoop *ofpk;
    knoop *nfpk;
    knoop *hpk;

    ofpk = *ppk;           /* 1st knoop */
    nfpk = (*ppk)->next; /* 2de knoop */
    hpk = (*ppk)->next; /* 2de knoop */

    while(hpk->next!=0){      /* voorlaaste knoop zoeken (niet de nullpointerknoop) */
        hpk = hpk->next;
    }

    hpk->next = ofpk;       /* voorlaaste knoop = ofpk */
    ofpk->next = 0;          /* de laaste knoop = nullpointer */
    *ppk = nfpk;             /* verzetten van de eerste knoop */
}
```

Examen C (15/01/2020) (versie 2)

Vraag 2)

```
int main() {
    int g = 013323;
    int c = 03;
    int res;

    printf("Before deleting\n");
    printf("g is = %o\n",g);

    printf("After deleting\n");
    res = verwijder_octaal_cijfer(g,c);
    printf("g is = %o",res);

    return 0;
}

/* -----
int verwijder_octaal_cijfer(int g,int c){
    int res;
    int octaalC;
    int verwijderC;

    if(g!=0){
        res = verwijder_octaal_cijfer(g>>3,c);
        octaalC = g&07;
        verwijderC = c^(g&07);
        if(verwijderC!=0){
            res = res<<3;
            res = res | octaalC;
        }
        return res;
    }
    else{
        return 0;
    }
}
```

Vraag 4)

```
int main(){
    char cs_d[] = "pbrofdicieagth";
    char cs_s[] = "bdeghj";

    puts(cs_d);
    verwijder_letters(cs_d,cs_s);
    printf("\n");
    puts(cs_d);

    return 0;
}

/*
void verwijder_letters(char *pacs_d, char *pacs_s){
    int an[26]={0};
    char *hpacs= pacs_d;

    while(*pacs_s!=0){
        an[*pacs_s-'a']=1;
        pacs_s++;
    }

    while(*hpacs!=0){
        if(an[*hpacs-'a']==0){
            *pacs_d= *hpacs;
            pacs_d++;
        }
        hpacs++;
    }
    *pacs_d=0;
}
```

Vraag 5)

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <stdbool.h>
#define MAX_A_C 80
#define SIZE_ARRAY 15

typedef struct {
    char *naam;
    int leeftijd;
} Persoon;

Persoon fillStruct( char *, int );
void printStruct(const Persoon *);
void lees_en_voeg_toe(Persoon *, int *);

int main() {
    int i;
    int g = 4;

    Persoon * personen ;
    personen = (Persoon *) malloc(SIZE_ARRAY * sizeof(Persoon *));

    personen[0] = fillStruct("Adam", 20);
    personen[1] = fillStruct("Belle", 16);
    personen[2] = fillStruct("Ce", 11);
    personen[3] = fillStruct("Dirk", 21);

    printf("De initiele personen zijn: \n");
    for(i = 0 ; i<g ; i++) {
        printStruct(&personen[i]);
    }

    lees_en_voeg_toe(personen,&g);

    printf("\n\nArray bevat nu %d elementen: \n", g);

    for(i = 0 ; i<g ; i++) {
        printStruct(&personen[i]);
    }

    for(i = 0 ; i<g ; i++) {
        free(personen[i].naam);
    }

    return 0;
}
```

```

/*-----*/
void printStruct(const Persoon * pst_c) {
    printf("%s is %d jaar.\n", pst_c->naam, pst_c->leeftijd);
}

/*-----*/
void lees_en_voeg_toe(Persoon *personen, int *pn) {

    char * naam;
    int leeftijd;

    printf("\nGeef naam: ");
    fgets(naam,MAX_A_C + 1,stdin);
    printf("Geef leeftijd: ");
    scanf("%d", &leeftijd);
    int len_naam = strlen(naam);
    if(naam[len_naam-1]== '\n') {
        naam[len_naam-1] = 0;

    }

    printf("Nieuwe persoon heeft naam %s en leeftijd %d", naam, leeftijd);

    /* Nieuwe persoon aanmaken */
    Persoon persoon;
    persoon.naam = (char *) malloc( len_naam * sizeof(char));
    strcpy(persoon.naam, naam);
    persoon.leeftijd = leeftijd;

    bool done = false;
    int j = *pn ;

    j -= 1;
    personen[j+1] = personen[j];

    /*NOTE: normaal zou deze 'if' loop dus wel een 'while' moeten zijn */
}

```

```
if (done == false){
    if (j == 0 || strcmp(personen[j].naam, naam) < 0 ){
        personen[j] = persoon;
        done = true;
    } else if (strcmp(personen[j].naam, naam) == 0 ){
        if (personen[j].leeftijd > leeftijd ){
            personen[j+1] = personen[j];
        }
        personen[j] = persoon;
        done = true;
    }
    else {
        personen[j+1] = personen[j];
    }
    j--;
}
*pn += 1;

}
```

Vraag 6)

```
/* ----- */
void zet_laatste_knoop_eerst(knoop **ppk) {
    knoop *hpk1 = (*ppk);
    knoop *hpk2 = 0;

    while(hpk1->next!=0){
        hpk2 = hpk1;
        hpk1 = hpk1->next;
    }
    hpk2->next = 0;
    hpk1->next = (*ppk);
    (*ppk) = hpk1;
}
```

Examen C (.../01/2021)

Vraag 2)

```
#include <stdio.h>
#include <string.h>
#include <stdbool.h>
#include <math.h>
#define EPSILON 0.000001

void print_int(const int *);
void print_double(const double *);
void print_cs(const char* const* );
void printArrayVoid(const void *, int, int, char, void (*) (const void*));

int cmpInt(const void *, const void * );
int cmpFloat(const void *, const void * );
int cmpCS(const void *, const void * );

void* getArrayExtr_rec(const void *, int, int, int(*)(const void*,const void*));

int main() {

    int an[] = {21,30,18,14,22,19};
    int n1 = sizeof(an)/sizeof(int);
    int gr1 = sizeof(int);

    double ad[] = {0.5,1.6,8.2,-2.4};
    int n2 = sizeof(ad)/sizeof(double);
    int gr2 = sizeof(double);

    char* apcs[] = {"Tom","Piet","Pol","Luc","Bram"};
    int n3 = sizeof(apcs)/sizeof(char* );
    int gr3 = sizeof(char* );

    void *pv1;
    void *pv2;
    void *pv3;

    printArrayVoid(an,n1,gr1,',',(void (*) (const void* ))print_int);
    printArrayVoid(ad,n2,gr2,'~',(void (*) (const void* ))print_double);
    printArrayVoid(apcs,n3,gr3,';',(void (*) (const void* ))print_cs);

    pv1 = getArrayExtr_rec(an,n1,gr1,cmpInt);
    pv2 = getArrayExtr_rec(ad,n2,gr2,cmpFloat);
    pv3 = getArrayExtr_rec(apcs,n3,gr3,cmpCS);

    printf("\n\n max is %d", *(int*)pv1);
    printf("\n max is %.2f", *(double*)pv2);
    printf("\n max is %s", *(char**)pv3);

    return 0;
}
```

```

/*
-----*/
void print_int(const int *pn_c){
    printf("%d",*pn_c);
}

void print_double(const double *pd_c){
    printf("%.2f",*pd_c);
}

void print_cs(const char* const* c_papcs_c){ /* !! DUBBEL STER !! */
    printf("%s",*c_papcs_c);
}

void printArrayVoid(const void *pav_c, int n, int gr, char tussenteken, void (*pfun)(const void*)){
    const char* hpc_c = (const char*)pav_c;
    int i;
    pfun(hpc_c);
    for(i=1; i < n; i++){
        printf("%c",tussenteken);
        hpc_c += gr;
        pfun(hpc_c);
    }
    printf("\n\n");
}

/*
-----*/
int cmpInt(const void *p1, const void *p2) {
    const int *i1 = p1;
    const int *i2 = p2;
    return (*i1 > *i2) - (*i1 < *i2);
}

int cmpFloat(const void *p1, const void *p2) {
    const float *f1 = p1;
    const float *f2 = p2;
    return (*f1 > *f2) - (*f1 < *f2);
}

int cmpCS(const void * a, const void * b){
    char *cs1 = *(char **)a;
    char *cs2 = *(char **)b;

    int rc = 0;
    if (strcmp(cs1, cs2) > 0)
        rc = 1;
    return rc;
}

void* getArrayExt_rec(const void *pav_c, int n, int gr, int (*pfun)(const void *, const void *)){
    char *pc = (char *)pav_c;
    void *max = pc;
    int i;
    for (i=1;i<n;i++){
        if (pfun(&pc[i*gr],max) > 0)
            max = &pc[i*gr];
    }
    return max;
}

```

Vraag 3)

```
/* ----- */
void wissel_knopen(knoop **ppk, int n) {
    knoop *hpk1;
    knoop *hpk2;
    knoop *hpk3;
    knoop *hpk4;
    int i=1;

    hpk1 = (*ppk);

    while(hpk1->next!=0 && i!=n){
        hpk2 = hpk1;          /* storing: i-1 knoop */
        hpk1 = hpk1->next;   /* storing: i knoop */
        i++;
    }
    hpk3 = hpk1->next;        /* storing: i+1 knoop */
    hpk4 = hpk3->next;        /* storing: i+2 knoop */

    hpk2->next = hpk3;
    hpk3->next = hpk1;
    hpk1->next = hpk4;

}
```

Vraag 4)

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <stdbool.h>
#include <math.h>
#define SIZE_ARRAY 4
#define EPSILON 0.000001

typedef struct{
    double lengte;
    double breedte;
}rechthoek;

typedef struct{
    rechthoek r;
    char kleur[50];
}gekleurdeRechthoek;

void verplaats_ptr(const gekleurdeRechthoek**, int);

/*
int main() {
    int g = 4;
    const gekleurdeRechthoek *grC;
    gekleurdeRechthoek *gr;
    gr = (gekleurdeRechthoek*)malloc(SIZE_ARRAY * (sizeof(gekleurdeRechthoek)));

    char *kl1 = "rood";
    char *kl2 = "oranje";
    char *kl3 = "geel";
    char *kl4 = "groen";

    strncpy(gr[0].kleur,kl1,50);
    gr[0].r.lengte = 1.0;
    gr[0].r.breedte= 2.0;
    printf("gekleurdeRechthoek1 \nLengte= %.1f \nBreedte= %.1f \nKleur= %s\n\n", gr[0].r.lengte,
    | gr[0].r.breedte, gr[0].kleur);

    strncpy(gr[1].kleur,kl2,50);
    gr[1].r.lengte = 1.0;
    gr[1].r.breedte= 2.0;
    printf("gekleurdeRechthoek2 \nLengte= %.1f \nBreedte= %.1f \nKleur= %s\n\n", gr[1].r.lengte,
    | gr[1].r.breedte, gr[1].kleur);

    strncpy(gr[2].kleur,kl3,50);
    gr[2].r.lengte = 2.0;
    gr[2].r.breedte= 2.0;
    printf("gekleurdeRechthoek3 \nLengte= %.1f \nBreedte= %.1f \nKleur= %s\n\n", gr[2].r.lengte,
    | gr[2].r.breedte, gr[2].kleur);
}
```

```

strncpy(gr[3].kleur,k14,50);
gr[3].r.lengte = 1.0;
gr[3].r.breedte= 5.0;
printf("gekleurdeRechthoek4 \nLengte= %.1f \nBreedte= %.1f \nKleur= %s\n\n", gr[3].r.lengte,
gr[3].r.breedte, gr[3].kleur);

grC=gr;
printf("\n\ngekleurdeRechthoek \nLengte= %.1f \nBreedte= %.1f \nKleur= %s\n\n", gr[0].r.lengte,
| gr[0].r.breedte, gr[0].kleur);
verplaats_ptr(&grC,g);
printf("\n\ngekleurdeRechthoek \nLengte= %.1f \nBreedte= %.1f \nKleur= %s\n\n", grC[0].r.lengte,
| grC[0].r.breedte, grC[0].kleur);

free(gr);
return 0;
}

/*
void verplaats_ptr(const gekleurdeRechthoek **past, int n) {
int i=0;
char cs[51];
int scf;
printf("Geef een kleur in: ");
scf = scanf("%50s",cs);

while(scf==0){
    fflush(stdin);

    printf("Geef een kleur in: ");
    scf = scanf("%50s",cs);
}
while(i<n && strcmp((*past)->kleur,cs)!=0 && fabs((*past)->r.lengte - (*past)->r.breedte)>EPSILON ){
    printf("%s \n", (*past)->kleur);
    (*past)++;
    i++;
}
}

```

Vraag 5)

```
int main() {
    if(controleerDubbelOctaal(067653)){
        printf("Wel dubbels");
    }
    else{
        printf("Geen dubbels");
    }
    return 0;
}

/*
-----*/
int controleerDubbelOctaal(int n){
    int b1=0;
    int b2=0;

    if((n>>3)==0) {
        return 0;
    }
    b1 = n&07;           /* Opslaan van het minst signifante cijfer */
    b2 = (n>>3)&07;     /* Opslaan van het volgende minst signifante cijfer */
    if((b1^b2)== 0) {    /* Als b1 en b2 gelijk zijn => b1^b2 == 0 */
        return 1;
    }
    controleerDubbelOctaal(n>>3);
}
```

Vraag 6)

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
char* encode(const char *, int);

int main(){
    char cs[]="Hallo!";
    char *pcs;

    printf("cs= %s\n", cs);
    pcs =encode(cs,2);
    printf("cs= %s",pcs);

    return 0;
}

char* encode(const char *pcs, int k){
    /* Data storage */
    /* const char *pcs */      /* (1) C-string          */
    int sizeMa;                /* (2) MAX size of C-string /* incl. null char */
    int sizeMi;                /* (3) MIN size of C-string /* excl. null char */
    int sizeDT;                /* (4) Size of data type   */
    int sizeH;                 /* (5) Size of heap        */
    char *hpcs;                /* (6) CharPointer voor geheugen op te slaan op de heap*/
    int i;

    sizeMa = strlen(pcs);
    sizeMi = sizeMa;
    sizeDT = sizeof(char);
    sizeH = (sizeMi+1)*sizeDT;
    hpcs = malloc(sizeH);

    for(i=0; i<sizeMa; i++){
        if(isupper(pcs[i])){
            hpcs[i]=tolower(pcs[i]);
            hpcs[i]=hpcs[i]+k;
        }
        else if(!isalpha(pcs[i])){
            hpcs[i]= pcs[i];
        }
        else{
            hpcs[i]=pcs[i]+k;
        }
    }
    return hpcs;
}
```


C++

Examenvragen opl

Master of Science in de Industriële
Wetenschappen

Industrieel Ingenieur



Examen C++ (xx/01/2016) (versie 1)

Vraag 1)

```
#include <iostream>
#include <map>
#include <vector>
#include <set>
#include "containers.h"
using namespace std;

/*-----*/
template <typename T, typename S>
void verwijder(const map<T, set<S>> &mp, vector<S> &v){

    typename map<T, set<S>> :: const_iterator itM;
    typename vector<S>:: iterator itV;

    itM = mp.begin();
    itV = v.begin();
    while(itV != v.end()){
        while(itM != mp.end()){
            auto itS= (itM->second).begin();
            while(itS != (itM->second).end()){
                if((*itV)==(*itS)){
                    cout << "Deleted" << endl;
                    v.erase(itV);
                }
                itS++;
            }
            itM++;
        }
        itV++;
    }
}
```

```

int main(){
    map<string, set<int>> mp;
    set<int> st1;
    set<int> st2;
    vector <int> v;

    st1.insert(1);
    st1.insert(2);
    st1.insert(3);
    st1.insert(4);
    st1.insert(5);

    st2.insert(6);
    st2.insert(7);
    st2.insert(8);
    st2.insert(9);
    st2.insert(10);

    mp.insert(pair<string, set<int>>("A",st1));
    mp.insert(pair<string, set<int>>("B",st2));

    v.push_back(3);
    v.push_back(8);
    v.push_back(16);
    v.push_back(11);

    cout << "\n-----\n" << endl;
    cout << "Map:" << endl;
    cout << "sizeMap: " << mp.size() << endl;
    cout << mp;

    cout << "\n-----\n" << endl;
    cout << "Vector:" << endl;
    cout << "sizeVector: " << v.size() << endl;
    cout << v;

    cout << "\n-----\n" << endl;
    verwijder(mp,v);

    cout << "\n-----\n" << endl;
    cout << "Map:" << endl;
    cout << "sizeMap: " << mp.size() << endl;
    cout << mp;

    cout << "\n-----\n" << endl;
    cout << "Vector:" << endl;
    cout << "sizeVector: " << v.size() << endl;
    cout << v;

    return 0;
}

```

Vraag 2)

```
#include <functional>
#include <fstream>
#include <map>
#include <iostream>
#include <stdexcept>
#include <string>
using namespace std;

string geef_eerste_regel(const string &bn, function<bool(const string &)>f){
    ifstream inv;
    inv.open(bn);
    if(!inv.is_open()){
        throw "Emessage: Failed to open file";
    }
    else{
        string s;
        getline(inv,s);

        while(!inv.fail()){
            if(f(s)==true){
                return s;
            }
            getline(inv,s);
        }
    }
    if(inv.fail()){
        throw logic_error("Bestand bevat geen woorden.");
    }
}

/*
int main(){
    try{
        string s= geef_eerste_regel("verhaal.txt",[](const string &regel){
            if(regel.find("verjaardag") != string::npos){
                return true;
            }
            else{
                return false;
            }
        });
        cout<< "Regel: " << s<< endl;
    }
    catch(const char *cs){
        cout<< cs << endl;
    }
    catch(const logic_error &e){
        cout<< "Exceptie: " << e.what();
    }

    return 0;
}

```

Examen C++ (xx/01/2016) (versie 2)

Vraag 2)

```
#include <iostream>
#include <map>
using namespace std;

template <typename T, typename S>
void keer_om(const map<T,S> &, map<S,T>&);

template <typename T, typename S>
ostream& operator<< (ostream &, map<T,S> &);

int main(){
    map<string, int> mp;
    map<int, string> omp;
    map<string, int>::iterator it;

    mp["Key"] = 1;

    mp["jan"] = 10;
    mp["an"] = 20;
    mp.insert(pair<string,int>("piet",30));

    cout << "\n-----\n" << endl;

    keer_om(mp,omp);

    cout <<"Map:" << endl;
    cout << "sizeMap: " << mp.size() << endl;
    cout << mp;

    cout << "\n-----\n" << endl;

    cout <<"Omap:" << endl;
    cout << "sizeOMap: " << omp.size() << endl;
    cout << omp;

    return 0;
}

/*
template <typename T, typename S>
// < K ; V >
void keer_om(const map<T,S> &mp, map<S,T> &omp){
    typename map<T,S>:: const_iterator it;
    it = mp.begin();

    while(it != mp.end()){
        omp[it->second] = it->first;
        it++;
    }
}
```

Vraag 3)

```
/*
vector<int> is_even(const string &s, int n, function<bool(int)>fun){

    ifstream inv(s);

    if(inv.is_open()){

        // Stack opvullen
        stack<int> st;
        int num;
        inv >> num;

        while(!inv.fail()){
            if(fun(num)){
                st.push(num);
            }
            inv >> num;
        }

        // Gegevens van de stack opvragen & vector opvullen
        int i=0;
        vector<int> v;

        while(i<n && i<st.size()){
            v.push_back(st.top());
            st.pop();
        }
        return v;
    }
    else{
        // Exceptie opwerpen (Standaardstring)
        throw "EMessage: Openen bestand " + s + " mislukt";
    }
}

int main() {
    try {
        vector<int> v = is_even("Oef3.txt", 6, [] (int x) {
            return (x%2) == 0;
        });
        cout << v;

    } catch (const string &Emessage) {
        cout << Emessage;
    }

    return 0;
}
```

Examen C++ (xx/01/2017) (versie 1)

Vraag 2)

```
vector<int> geef_getallen(const string &bn, int n,function<bool(const int &)>f){
    ifstream inv;
    inv.open(bn);
    if(!inv.is_open()){
        throw "Emessage: Failed to open inFile " + bn;
    }
    else{
        stack<int> st;
        int getal;
        inv>> getal;
        while(!inv.fail()){
            if(f(getal)==true){
                st.push(getal);
            }
            inv>>getal;
        }
        if(st.size()<n){
            throw "Emessage: Bestand bevat te weinig getallen";
        }
        vector<int> v;
        int i=0;
        while(i<n){
            v.push_back(st.top());
            st.pop();
            i++;
        }
        return v;
    }
    if(inv.fail()){
        throw logic_error("Bestand bevat geen getallen");
    }
}
```

```

int main(int argc, char**argv){
    if (argc == 1){
        cout << "ErrorMessage: There are no strings on the command line";
    }
    if(argc>3){
        cout << "ErrorMessage: There are to many strings on the command line";
    }
    else{
        try{
            cout<< "There are " << argc << " files on the command line" << endl<< endl;
            cout<< "These are those files: "<<endl;
            for(int i=0;i<argc;i++){
                cout<< argv[i] << " , ";
            }
            cout << endl<< endl;
            string inFile = argv[1];
            string outFile = argv[2];
            vector<int> v = geef_getallen(inFile,10,[](const int &getal){
                if((getal>0) && (getal%3==0) || (getal%5==0)){
                    return true;
                }
                else{
                    return false;
                }
            });
            cout<< v << endl;
            ofstream uitv;
            uitv.open(outFile, ios::app);
            if(uitv.fail()){
                cout << "ErrorMessage: Failed to open outFile." << endl;
                exit(1);
            }

            for(int i=0;i<v.size();i++){
                uitv<< v[i] << " ";
            }
            uitv << " "<< endl;
        }
        catch(const string &s){
            cout<< s << endl;
        }
        catch(const char *cs){
            cout<< cs << endl;
        }
        catch(logic_error &e){
            cout<< "Emessage: " << e.what();
        }
    }
    return 0;
}

```

Vraag 3)

```
class A {
    public:
        A(int _a = 9);
        friend ostream& operator<<(ostream &, const A&);
    private:
        int a;

    protected:
        virtual void schrijf(ostream&) const;
};

class B : public A {
    public:
        B(int _a = 4, int _b = 5);
        void schrijf(ostream& os) const;

    private:
        int b;
};

int main() {
    A a;
    A a2(5);
    B b;
    B b2(1);
    B b3(2, 3);
    cout << "a:" << endl << a << endl;
    cout << "a2:" << endl << a2 << endl;
    cout << "b:" << endl << b << endl;
    cout << "b2:" << endl << b2 << endl;
    cout << "b3:" << endl << b3 << endl;

    return 0;
}

A::A(int _a) : a(_a) {

void A::schrijf(ostream& os) const {
    os << "a= " << a << endl;
}

ostream& operator<<(ostream& os, const A& a) {
    a.schrijf(os);
    return os;
}

B::B(int _a, int _b) : A(_a), b(_b) {
```

```
void B::schrijf(ostream& os) const {
    A::schrijf(os);
    os << "b= " << b << endl;
}
```

Vraag 4)

```
#include <iostream>
#include <map>
using namespace std;

template <typename T, typename S>
void keer_om(const map<T,S> &, map<S,T>&);

template <typename T, typename S>
ostream& operator<< (ostream &, map<T,S> &);

int main(){
    map<string, int> mp;
    map<int, string> omp;
    map<string, int>::iterator it;

    mp["Key"] = 1;

    mp["jan"] = 10;
    mp["an"] = 20;
    mp.insert(pair<string,int>("piet",30));

    cout << "\n-----\n" << endl;

    keer_om(mp,omp);

    cout <<"Map:" << endl;
    cout << "sizeMap: " << mp.size() << endl;
    cout << mp;

    cout << "\n-----\n" << endl;

    cout <<"Omap:" << endl;
    cout << "sizeOMap: " << omp.size() << endl;
    cout << omp;

    return 0;
}

/*
template <typename T, typename S>
// < K ; V >
void keer_om(const map<T,S> &mp, map<S,T> &omp){
    typename map<T,S>:: const_iterator it;
    it = mp.begin();

    while(it != mp.end()){
        omp[it->second] = it->first;
        it++;
    }
}
```

Examen C++ (xx/01/2020)

Vraag 2)

```
/*-----*/
vector<string> controleer_woorden(const string &bn, int n, function<bool(const string &)>f){
    ifstream inv;
    inv.open(bn);
    if(!inv.is_open()){
        throw "Emessage: Failed to open file";
    }
    else{
        string woord;
        stack<string> st;
        inv>> woord;

        while(!inv.fail()){
            if(f(woord)==true){
                st.push(woord);
            }
            inv>> woord;
        }
        vector<string> v;
        v.reserve(n);
        int i=0;
        while (i<n) {
            v.push_back(st.top());
            st.pop();
            i++;
        }
        return v;
    }
    if(inv.fail()){
        throw logic_error("Bestand bevat geen woorden");
    }
}

/*-----*/
template<typename T>
ostream& operator<<(ostream &out, const vector<T> &v){
    for(int i=0;i<v.size();i++){
        cout << v[i] << " ";
    }
    cout << endl;
    return out;
}
```

```

/*
int main() {
try{
    vector<string> v;
    v = controleer_woorden("verhaal.txt",5, [](const string &woord){
        int n= woord.length();
        if((woord[0]>= 'a' && woord[0]<='z') && (woord[n-1]== '?') && (woord!="ex")){
            return true;
        }
        else{
            return false;
        }
    });
    cout<< "VWD correct " << v;
}
catch(const char *cs){
    cout<< cs << endl;
}
catch(const logic_error &e){
    cout<< "Exception: " << e.what();
}

return 0;
}

```

Vraag 4)

```
set<string> geef_voornaam(const string &familienaam, const map<int, map<string, set<string>>>&personen ){
    set<string> stVoornamen;
    set<string> stLeeg;
    int n=0;
    for(const auto &pers: personen){ // (1ste) map overlopen
        auto it_find= (pers.second).find(familienaam);
        if(it_find!=(pers.second).end()){ // (2de) map overlopen
            auto it = it_find->second.begin();
            if(it != it_find->second.end()){ // set overlopen
                stVoornamen.insert(*it);
                n++;
            }
            it++;
        }
        it_find++;
    }
    if(n>5){
        return stVoornamen;
    }
    else{
        return stLeeg;
    }
}
```

Examen C++ (xx/01/2021)

Vraag 1)

```
class Langeslang : public vector<unique_ptr<int>>{

public:
    // Default constructor

    // (Destructor)

    // Parameter constructor

    // (Copy constructor)

    // Member functions
    void vul(const vector<int>& v);
    Langeslang& concatenate(Langeslang &);

    // Friend function
    friend ostream& operator << (ostream& , const Langeslang& );

private:
    // Const member function
    void schrijf(ostream& )const;

};

int main(){
    Langeslang a;
    Langeslang b;
    Langeslang c;

    a.vul({1,2});      // via a vind je de getallen 1 en 2
    b.vul({3,4,5});    // via b vind je de getallen 3, 4 en 5
    c.vul({6,7});      // via c vind je de getallen 6 en 7

    cout << "a: " << a << endl << "b: " << b << endl << "c: " << c << endl << endl;

    a.concatenate(a);
    // via a vind je nu de 4 getallen 1, 2, 1 en 2

    cout << "na a.concatenate(a)" << endl;
    cout << "a: " << a << endl << "b: " << b << endl << "c: " << c << endl << endl;

    a.concatenate(b).concatenate(c);
    // via a vind je nu de 9 getallen 1, 2, 1, 2, 3, 4, 5, 6 en 7
    // b is leeg
    // c is leeg
```

```

cout << "na a.concatenate(b).concatenate(c)" << endl;
cout << "a: " << a << endl << "b: " << b << endl << "c: " << c << endl << endl;
return 0;
}

/*
// Member functions
Langeslang& Langeslang::concatenate(Langeslang &cObject){
    int sizeObject = size();

    resize(sizeObject + cObject.size());

    if(this == &cObject)
    {
        for(int i=0; i<sizeObject; i++){
            (*this)[i+sizeObject] = make_unique<int>(*cObject[i]);
        }
    }

    else{
        for(int i=sizeObject; i<size(); i++)
        {
            (*this)[i] = move(cObject[i-sizeObject]);
        }
        cObject.resize(0);
    }
    return *this;
}

/*
void Langeslang::vul(const vector<int> &v){
    int sizeVObject = v.size();

    resize(sizeVObject);

    for(int i=0; i<size(); i++)
    {
        (*this)[i] = make_unique<int>(v[i]);
    }
}

```

Vraag 2)

```
string alfabetisch_kleinste(const string &bn, int n,function<bool(const string &)>f){
    ifstream inv;
    inv.open(bn);

    if(!inv.is_open()){
        throw "Emessage: Failed to open file";
    }
    else{
        cout<< "Opening File: " + bn << endl;
        map<string,int> mp;
        map<string,int>::iterator it;

        string woord;
        string akwoord;
        inv>> woord;

        mp.insert(pair<string,int>(woord,1));

        while(!inv.fail()){

            if(mp.find(woord)==mp.end()){
                mp[woord]=1;
            }
            else{
                mp[woord]++;
            }
            inv>> woord;
        }

        for (auto& it : mp) {
            cout << it.first << "-"<< it.second << endl;
        }

        it = mp.begin();
        while (it != mp.end()) {
            if( f(it->first)==true && (it->second)>=n){
                return it->first;
            }
            it++;
        }
    }

    if(inv.fail()){
        throw logic_error("Bestand bevat geen woorden");
    }
}
```

```
/*
int main(){
    try{
        string s = alfabetisch_kleinste("verhaal.txt",5,[](const string &woord){
            int n = woord.length();
            if((woord[0]==woord[n-1]) && woord.find('p')== string::npos){
                return true;
            }
            else{
                return false;
            }
        });

        cout << "Alfabetisch kleinste woord uit file: " << s << endl;
    }
    catch(const char *cs){
        cout<< cs << endl;
    }
    catch(const logic_error &e){
        cout<< "Exception: " << e.what();
    }
    return 0;
}
```

Vraag 3)

```
class Basis{
public:
    // Default constructor

    // Parameter constructor
    Basis(int =1);

    // Copy constructor

    // Member function
    bool operator==(const Basis&) const;

    // Friend functie
    friend ostream& operator<<(ostream &,const Basis &);

protected:
    int getal;
    virtual void print(ostream &)const;
};

class Afgeleid :public Basis{
public:
    // Default constructor

    // Parameter constructor
    Afgeleid(int = -2, int = 4);

    // Copy constructor

    // Member function
    bool operator==(const Afgeleid &)const;

private:
    int extra;
    virtual void print(ostream &)const;
};

};
```

```

/*
int main(){

    Basis a;      // getal= 1
    Basis b(3);  // getal= 3
    Basis c(b);
    if(b==c){
        cout<< b << " == " <<c << endl;
    }

    Afgeleid d;      // getal= -2, extra=4
    Afgeleid e(9);   // getal= 9, extra=4
    Afgeleid f(9,7); // getal= 9, extra=7
    Afgeleid g(f);
    if(e==f){
        cout<< "af " << e << " == " <<f;
    }
    return 0;
}

// Class_Basis (Base class)
Basis::Basis(int x): getal(x){

}
bool Basis::operator==(const Basis &r) const{
    bool b= false;
    if((getal==r.getal)){
        b=true;
    }
    return b;
}

void Basis::print(ostream &out) const {
    out<<"getal: " <<getal;
}

ostream& operator<<(ostream &out, const Basis &r){
    r.print(out);
    return out;
}

```

```
//////////////////////////////  
// Class_Afgeleid (Derived class)  
  
Afgeleid::Afgeleid(int x, int y): Basis(x), extra(y){  
}  
  
bool Afgeleid::operator==(const Afgeleid &r) const{  
    bool b= false;  
    if((getal==r.getal) && (extra==r.extra)){  
        b=true;  
    }  
    return b;  
}  
  
void Afgeleid::print(ostream &out) const{  
    Basis::print(out);  
    out<< ", extra:" << extra;  
}
```

Vraag 4)

```
int tel_kranten(const map<string, map<string, set<int>>> straten &straten,
const string &kranten){
    int tel=0;
    for(const auto &straat:straten){
        auto it = (straat.second).find(krant);
        if(it != (straat.second).end()){
            tel+= (straat.second).size();
        }
    }
    return tel;
}
```