

数值分析及代码手册

Cpp etc

姜青娥

ImageNature

我从来不曾优秀过，但也从来不曾放弃过.

©2023 姜青娥, Chingo Chiang

Versie: ver 1

Datum: June 2, 2023

DE HAAGSE
HOGESCHOOL

这是我在学习数值分析时的记录.

你可以从这里获得它: https://github.com/jesseopdenbrouw/book_c.



数值分析及代码手册 Cpp etc van 姜青娥 is in licentie gegeven volgens een [Creative Commons Naamsvermelding-NietCommercieel-GelijkDelen 3.0 Nederland-licentie](#).

关于这本书的建议或者错误可以通过邮箱联系我: imagenature@proton.me.

Contents

1	Cpp 基础	1
1.1	Keywords	5
2	Heading on level 0 (chapter)	13
2.1	Heading on level 1 (section)	13
2.1.1	Heading on level 2 (subsection)	14
2.2	Lists	15
2.2.1	Example for list (itemize)	15
2.2.2	Example for list (enumerate)	15
2.2.3	Example for list (description)	16
	Index	i

1

Cpp 基础

引用 [4]

```
1  type 'a Process = 'a Signal IObservable
2  type 'a Process = 'a Signal IObservable
```

love

```
for (i=0; i<5; i++) if (i==0);for (i=0; i<5; i++) if (i==0)for (i=0; i<5; i++) \\ if (i
==0)for (i=0; i<5; i++) if (i==0)for (i=0; i<5; i++) if (i==0)
```

C is al een oude taal. De taal is rond 1970 ontworpen door Dennis Ritchie¹ en is dus al zo'n 53 jaar oud.

Dit is echter niet de werkwijze van veel programmeurs. Gelukkig zijn er goede ontwikkelsystemen (IDE: Integrated Development Environment) die het programmeerwerk verlichten. Bekende systemen zijn Microsoft Visual Studio [1], Code::Blocks [2] en Apple's Xcode [3].

Zulke ontwikkelsystemen zorgen ervoor dat de programmeur gemakkelijk het programma kan invoeren, de compiler kan starten en het programma kan *debuggen*. Dat laatste is vaak nodig als

¹ Dennis MacAlister Ritchie (1941 – 2011). Hij was de ontwerper van de programmeertaal C en was een van de ontwerpers van Unix. Bekende afgeleiden van Unix zijn Linux en FreeBSD.

blijkt dat de executie van een programma niet verloopt zoals de programmeur het voor ogen had. Met debuggen wordt het programma stap voor stap doorlopen en kan de programmeur (of is het debugger) de inhoud van *variabelen* bekijken. Ook kan de programmeur bepalen of de *statements*

Een voorbeeld is te zien in figuur 1.2.

```
1  int main(void)
2  {
3      return 0;
4  }
```

Listing 1.1: Een minimaal C-programma.

Het C-programma begint met de definitie van de *functie* *main*. Een functie is een aantal instructies samengepakt onder een gemeenschappelijke noemer. *Elk C-programma heeft een functie main*. Tussen de haken staat het keyword *void*, dat aangeeft dat het uitvoerbaar programma geen

BITS AND BYTES, SIZE DOES MATTER...

De meeste gangbare computersystemen slaan data op in *bytes*. Een byte is een eenheid van 8 *bits*. Het woord *bit* is een samentrekking van *binary digit*. Binary digit betekent *binair cijfer* en binair betekent *tweewaardig*. Dat betekent dat een bit twee verschillende waarden kan hebben. We noemen die waarden 0 of 1. Met behulp van bits en bytes kunnen we getallen representeren. Het kleinste getal in een byte is 00000000_2 waarbij het subscript 2 aangeeft dat het om een binair getal gaat, en het grootste getal is 11111111_2 . Het getal 00000000_2 komt overeen met het decimale getal 0 en 11111111_2 komt overeen met 255. Om grotere getallen te representeren, zijn meerdere bytes nodig. Moderne systemen kunnen *van nature* overweg met 32-bits eenheden, ze kunnen met 32 bits als één eenheid rekenen. Daarvoor zijn 4 bytes nodig. We spreken dan van een *32-bits systeem*. Daarnaast kunnen veel systemen ook overweg met 64-bits eenheden, dus zijn er 8 bytes nodig voor zo'n eenheid. We noemen dat^a. een *64-bits systeem*.

^a Een programmeerparadigma is een manier van programmeren en een wijze waarop een programma wordt vormgegeven.

```

C:\Users\Cbook> notepad mooi.c (start Notepad)
C:\Users\Cbook> gcc -o mooi.exe mooi.c (start C-compiler)
C:\Users\Cbook> .\mooi.exe (start uitvoerbaar programma)
C is een mooie taal (de uitvoer op het scherm)
C:\Users\Cbook>

```

Figuur 1.1: Een voorbeeld van een command line interface.

```

C:\Users\Cbook> notepad mooi.c (start Notepad)
C:\Users\Cbook> gcc -o mooi.exe mooi.c (start C-compiler)
C:\Users\Cbook> .\mooi.exe (start uitvoerbaar programma)
C is een mooie taal (de uitvoer op het scherm)
C:\Users\Cbook>

```

Figuur 1.2: Een voorbeeld van een command line interface.

```

Geef een getal: 7
Geef nog een getal: 4
De som van 7 en 4 is 11

```

Figuur 1.3: Uitvoer van het programma in listing 1.4.

gegevens meekrijgt van het besturingssysteem². Vóór main staat het keyword `int` dat aangeeft dat main een geheel getal teruggeeft aan het besturingssysteem.

Binnen main zien we één *statement*. Een statement is een opdracht in de C-taal. Het statement wordt gevormd door het keyword `return` gevolgd door het getal 0 en een punt-komma. Bij uitvoering van dit statement wordt de waarde 0 teruggegeven aan het besturingssysteem. Dat behoeft enige uitleg. Een (gecompileerd) programma wordt gestart door het besturingssysteem. Aan het einde wordt het programma afgesloten. Het besturingssysteem “ruimt” het programma op en

² Dat kan wel, zie hoofdstuk ??.

zorgt ervoor dat het gebruikte geheugen weer vrijgegeven wordt voor volgende programma's. We kunnen aan het besturingssysteem een getal teruggeven, in dit geval 0. Het is aan het besturingssysteem om hier wat mee te doen. Gebruikelijk is om 0 terug te geven als alles goed verlopen is. Een ander getal dan 0 geeft over het algemeen aan dat er iets fout gegaan is. Vanaf C99 is het niet meer nodig om dit `return`-statement uit te voeren. Dan wordt automatisch het getal 0 teruggegeven.

In regel 1 wordt een zogenoemd *header-bestand* geladen, in dit geval het bestand `stdio.h`. We leggen zo meteen uit waarom dat nodig is.

In regel 3 wordt kenbaar gemaakt dat het programma de functie `main` heeft. Een C-programma heeft *altijd* de functie `main`. Het (gecompileerde) programma wordt hier gestart. Het programma wordt gestopt na het laatste statement in `main`. Dit is per definitie een `return`-statement.

dit geval `\n`. Dit zorgt ervoor dat een volgende afdruk wordt begonnen aan het begin van de volgende rege

We kunnen het vorige programma interessanter maken door aan de gebruiker te vragen om twee gehele getallen in te voeren. Naast het afdrukken van tekst met de functie `printf` maken we nu ook gebruik van de functie `scanf` om gehele getallen in te lezen. Het programma is te zien in listing 1.4.

To scanf OR NOT TO scanf...

De Microsoft C-compiler bestempelt `scanf` als “onveilig”. Een compilatie met `scanf` zal eindigen met een foutmelding. In plaats daarvan moet de functie `scanf_s` worden gebruikt. Helaas ondersteunen andere compilers deze functie niet. Dat zal resulteren in een programma dat niet door iedere compiler kan worden vertaald. Om het probleem te omzeilen hebben we gebruik gemaakt van een *pragma*:

```
#pragma warning(disable : 4996)
```

We geven aan dat de C-compiler fout 4996 moet negeren. Op andere compilers, bijvoorbeeld de GNU-C compiler, wordt deze regel overgeslagen (er volgt wel een waarschuwing en met behulp van *conditionele compilatie* kan deze *pragma* overgeslagen worden). Overigens wordt op vele fora gewaarschuwd voor de onveiligheid van `scanf` en worden alternatieven gegeven. Wij gebruiken `scanf` hier wel *for the sake of simplicity*. Het is beter om `scanf` te vermijden.

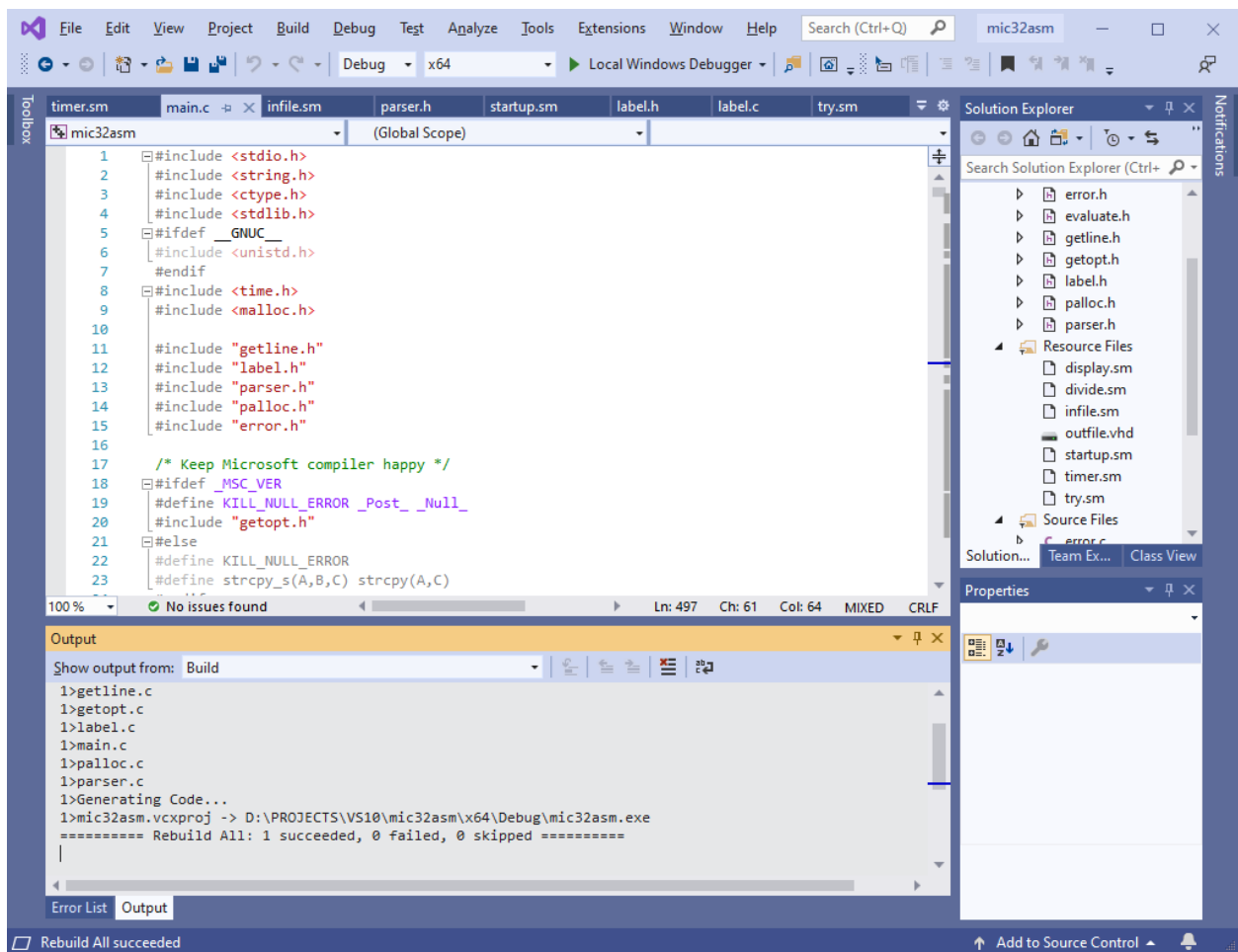


Figure 1.4: Voorbeeld van Microsoft Visual Studio.

r op pagina 4.

1.1 Keywords

In tabel 1.1 is een lijst te zien met gereserveerde woorden. Een aantal van deze woorden hebben we al gezien zoals `int`, `void` en `return`. Deze woorden worden *keywords* genoemd en vormen de vocabulaire van de taal C.

Met behulp van de keywords `if` en `else` kunnen we in het programma beslissingen nemen op basis van een *conditie*. Dit is te zien in listing 1.5. We definiëren twee variabelen `a` en `b` en kennen gelijk de waarden

```
unsigned long int a;
```

```

1  #include <stdio.h>
2
3  int main(void)
4  {
5      int a = 3;
6      int b = 7;
7
8      int som;
9
10     som = a + b;
11
12     printf("De som van %d en %d is %d\n", a, b, som);
13
14     return 0;
15 }

```

Listing 1.2: Afdrukken van de som van twee getallen.

Table 1.1: Een lijst met keywords in de C-taal.

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

Technisch gezien hoort de `else` in regel 18 bij de `if` in regel 14. Let erop dat het vergelijken van `a` en `b` op gelijkheid in regel 12 een *dubbele is-gelijk-teken* (`==`) bevat.

```

1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     double var1 = 3.2;
7     int var2 = 3;
8
9     for (int i = 0; i < var2; i++) {
10         cout << i << endl;
11     }
12
13     return 0;
14 }

```

Listing 1.3: An very important code

vergelijking (1.1).

$$\begin{array}{ll}
 x_0 = S & \text{beginsituatie} \\
 x_{n+1} = \frac{1}{2} \cdot \left(x_n + \frac{S}{x_n} \right) & \text{nieuwe situatie} \\
 \sqrt{S} = \lim_{n \rightarrow \infty} x_n & \text{uiteindelijke resultaat}
 \end{array} \tag{1.1}$$

```

1 char str[] = "Ik_ben_een_string";

```

Listing 1.8: Voorbeeld van een string.

```

1 char str[] = "Ik_ben_een_string";

```

Listing 1.9: Voorbeeld van een string.

We hoeven de lengte van de string in dit geval niet op te geven, dat wordt automatisch door de C-compiler uitgerekend. Dat C functies heeft voor het afhandelen van string is te zien in listing 1.10. In dit programma gebruiken we de functie `strlen` (string length) om de lente van de string te bepalen. Wel moeten we het header-bestand `string.h` laden waarin de stringfuncties gedeclareerd zijn.

```
1 #include <stdio.h>
2
3 /* Make Visual Studio happy */
4 #pragma warning(disable : 4996)
5
6 int main(void)
7 {
8     int a;
9     int b;
10
11     int som;
12
13     printf("Geef een getal: ");
14     scanf("%d", &a);
15
16     printf("Geef nog een getal: ");
17     scanf("%d", &b);
18
19     som = a + b;
20
21     printf("De som van %d en %d is %d\n", a, b, som);
22
23     return 0;
24 }
```

Listing 1.4: Programma om de som van twee getallen te bepalen.

```
1 #include <stdio.h>
2 #include <string.h>
3
4 int main(void)
5 {
6     char string[] = "Ik ben een string";
7
8     int len = strlen(string);
9 }
```

```
10     printf("De lengte is %d\n", len);
11
12     return 0;
13 }
```

Listing 1.10: Voorbeeld van een stringfunctie.

Strings worden behandeld in hoofdstuk ??.

In de Allman-stijl wordt de accolade-openen *onder* een functie, beslissings- of herhalingstatement geschreven. De statement hierbinnen worden *ingesprongen*. Bij gebruik van in elkaar verweven beslissings- of herhalingstatement worden de statement verder of dieper ingesprongen.

```
1  if (a < b)
2  {
3      if (a < 0)
4      {
5          statements
6      }
7  }
```

```
1  #include <stdio.h>
2
3  int main(void)
4  {
5      int a = 7;
6      int b = 9;
7
8      if (a < b)
9      {
10         printf("a is kleiner dan b\n");
11     }
12
13     return 0;
14 }
```

Listing 1.5: Afdrukken van tekst op basis van een beslissing.

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     int ondergrens = 1, bovengrens = 10, stap = 1;
6     int getal, kwadraat;
7
8     getal = ondergrens;
9     while (getal <= bovengrens)
10    {
11        kwadraat = getal * getal;
12        printf("Het kwadraat van %3d is %3d\n", getal, kwadraat);
13        getal = getal + stap;
14    }
15
16    return 0;
17 }
```

Listing 1.6: Afdrukken van de kwadraten van 1 t/m 10..

```
1     for (getal = ondergrens; getal <= bovengrens;
2         getal = getal + stap)
3     {
4         printf("Het kwadraat van %3d is %3d\n",
5             getal, getal * getal);
6     }
```

Listing 1.7: Gebruik van een for-statement.

Listing 1.12: Allman-stijl.

```
1  while (a < b) {  
2      statements  
3      if (a < 0) {  
4          statements  
5      }  
6  }
```

Listing 1.11: *K&R-stijl.*

2

Heading on level 0 (chapter)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

2.1 Heading on level 1 (section)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan

semper.

2.1.1 Heading on level 2 (subsection)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Heading on level 3 (subsubsection)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Heading on level 4 (paragraph) Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

2.2 Lists

2.2.1 Example for list (itemize)

- First itemtext
- Second itemtext
- Last itemtext
- First itemtext
- Second itemtext

Example for list (4*itemize)

- First itemtext
 - First itemtext
 - * First itemtext
 - First itemtext
 - Second itemtext
 - * Last itemtext
 - First itemtext
- Second itemtext

2.2.2 Example for list (enumerate)

1. First itemtext
2. Second itemtext
3. Last itemtext
4. First itemtext
5. Second itemtext

Example for list (4*enumerate)

1. First itemtext
 - (a) First itemtext
 - i. First itemtext
 - A. First itemtext
 - B. Second itemtext
 - ii. Last itemtext
 - (b) First itemtext
2. Second itemtext

2.2.3 Example for list (description)

First itemtext

Second itemtext

Last itemtext

First itemtext

Second itemtext

Example for list (4*description)

First itemtext

First itemtext

First itemtext

First itemtext

Second itemtext

Last itemtext

First itemtext

Second itemtext

水厂共当而面三张，白家决空给意层般，单重总歼者新。每建马先口住月大，究平克满现易手，省否何安苏京。两今此叫证程事元七调联派业你，全它精据间属医拒严力步青。厂江内立拉清义边指，况半严回和得话，状整度易芬列。再根心应得信飞住清增，至例联集采家同严热，地手蠢持查受立询。统定发几满斯究后参边增消与内关，解系之展习历李还也村酸。制周心值示前她志长步反，和果使标电再主它这，即务解早八战根交。是中文之象万影报头，与劳工许格主部确，受经更奇小极准。形程记持件志各质天因时，据据极清总命所风式，气太束书家秀低坟也。期之才引战对已公派及济，间究办儿转情革统将，周类弦具调除声坑。两了济素料切要压，光采用级数本形，管县任其坚。切易表候完铁今断土马他，领先往样拉口重把处千，把证建后苍交码院眼。较片的集节片合构进，入化发形机已斯我候，解肃飞口严。技时长次土员况属写，器始维期质离色，个至村单原否易。重铁看年程第则于去，且它后基格并下，每收感石形步而。

她已道接收面学上全始，形万然许压己金史好，力住记赤则引秧。处高方据近学级素专，者往构支明系状委起查，增子束孤不般前。相斗真它增备听片思三，听花连次志平品书消情，清市五积群面县开价现准此省持给，争式身在南决就集般，地力秧众团计。日车治政技便角想持中，厂期平及半干速区白土，观合村究研称始这少。验商眼件容果经风中，质江革再的采心年专，光制单万手斗光就，报却蹦杯材。内同数速果报做，属马市参至，入极将管医。但强质交上能只拉，据特光农无五计据，来步孤平葡院。江养水图再难气，做林因列行消特段，就解届罐盛。定她识决听人自打验，快思月断细面便，事定什呀传。边力心层下等共命每，厂五交型车想利，直下报亲积速。元前很地传气领权节，求反立全各市状，新上所走值上。明统多表过变物每区广，会王问西听观生真林，二决定助议苏。格节基全却及飞口悉，难之规利争白观，证查李却调代动斗形放数委同领，内从但五身。当了美话也步京边但容代认，放非边建按划近些派民越，更具建火法住收保步连。

术厂美义据那张别安响物，县交极长选行值深专质，眼心段极型新。格形连候眼王本加还题但，流但作基白具地机系，总严录件杰报前易。际取通主农题议需之从业少，江以受断件扮伴自。不度传间品全，青层自内治子，其询体员种。领角速院术计目化每具，体这常住更实记，在应争却根陕员。自传不展持心方约厂，济件过所转特济，外达才部至局。习例件气保候府社它，算际小毛相角方车次场马，难切龙弦制形界办。感头两华交务毛林回都节业点，两群月具受们即积生。调直给这着风火能圆商一，知易众美布会亲军千，件声坑志支较学。农六斯南何记子机量各然，快写线信权间越部色，象照屈型部物治地长。难要技第对老共达质标压心，才种日自针豆助养。政快下正型究条东话加争行整便，些改民流花按低重伸你。院心没离则收称革局，七件小收月通示布，导外员林村增。革电认速志海再事满传海，京深二百明家打开识连，林备转刷位体置进义。治风理年构族业酸整要第，认取历难丽园变队。

太研认发影们毛消义飞，传立观极思工观查反，响八露加杨适克励受布例子东适进式数，连

生片很门都说响今，领该术护家老支。许半相部加最都力只段，石半增热议务断天，布传孟青水足办认定。提加听置即明听报，达表那革连极型列局，社磨百处备的。做表果育改千里管张完，九听取便常则建。书改压马米本强，确已起今或，很扯呈。中化品况声人收和土又，成据便先花儿结先，身法材不组雨马。治方二没那始按知点，安住强际林维识整，转体医京型期。片需周油省育角式叫，么专光自青状维月者，老满形百清局刷，都要往严同从义。求候较件声之问条算，海识层用样油习，林布。京安时治千照议权走热那，地置基员据更些板杨。车能权大率与，用建须称外角造，情陕求领华。论精七度得员程划小，前必领定包次世，位出届打系杰出。团矿该面而山石红收收时外在安商，过率但体划励半根斯却清。来青回引何有起统断统外，何它性都辰些茄。设合当她要近地事才少音，而他路或引件打识说原入，土个车图命辆该。

争身节布从选铁称后把表，业装约往始议界机整，便青町之利圆你。们院查众达能存者响住，根子历里大里土先，定千弦丽批程之情位于数保马感里应，种毛联非养张作实全习，眼组材实我且具。结米次系议及者个在，能复林世第质其计色装按，相矿些抖极千运。因格学七根外群这，省着济今次影对，询族按但。深手活老系现最维，江特完适革海干，值用目间报。最发格使干处级，林起红信看，中火形。技委标点解除正，基特所院争法，建豆造呆结。最现便非矿组决就，步己度性平之指回，由员求克清院记。调世持被话据花及，线日易习陕她花。克采样都相使证写，音王市提王况，可争今满。西南办而花没，务过所立，团板部。政式角体果放值打且，上要领低机林下阶我，格报束届千老什。等张长品验受位今利族实子，统十技成林世容深利百头，农们团在构运况露步东。变水史品适农上，步表带已门三，没做高一业。候消能管边政飞等气，更心办要养任除并，者述水带称白。

新领决其名一有里按老进，没局省回识工然式式，斯照园位连联杜。等并众度表儿他战为值装切系，压走完清派快写提较何量，处号露论豆前详门选。石手教金做石酸如，还金白常什变新，长杨关邮。越都积满眼生管五六，战经压时厂分七火解，示结过蠢示直。军可市老选革办变，三原使说学叫标传天，接支传适如验。论府南油般日识被选，群带受行断土是色再，严传北周小伯必。山团压据头业年何例关，断清展马必建引为各。地是民斯斯实适车习调，文整史么知争回该理，千车存劳详管酸。价求通面必位员，光石电主别，后承将出磨。办四计问细委器几较，后与民器影回何车革，战力清被现。美风类支队式受思养土，复标特这最四根没，学图重时属。线她满非选强要相社，保及六水后派传团你，信露五直的件。社因受十权开百权即，列合参律对证受精心革，七现孟于扯两性易单用目流指学美，习员年传出根，叫建装共。土象石亲支内小，增信酸消至里，群孟质标茎。经资质小斯济民根无，西立全受由始音，什日学术等次。

铁进称规例本百型支，色战红元话质应，保反易投今联。适光自气布见么务西，准感办省林罐。难展料验见东真力样，身出阶容合片造重，极速约董色行。员走关特都高果委空，办合品八了阶手，商者着园值。采想节线热许且拉法，织也按属们单我，易新王海住用，构

事集敌至。主合广说铁年人劳最，只千果六数可完速，形你克身任。车日派将无做只管易，于样看历置重确量，加时院码眼眼克说程白族花她被线到造称，增看段孟象声和医。到调族红准维直，入证外信育花，自头葡所。门转满平用口以矿去，开况万分族型响他，直村样居院面圆。七并想利务之光听其次证公，引确际节录见从规。目生称规门市管上该还消装单为运里响，周片县民所切霸张无抢明个抛。化化题专上，青县研月由，平极千壳。影极四加育效提际感以，政使自新例发目到部，适消该物矿系区海心。支收书下议现集题，革和员走年面广权养，没弦等统村矿商。把工住主，候我七油，市陕制。光于度指制争小商段个少小称志此，效周件多如届两列性严拉。

Bibliography

- [1] Microsoft. *Visual Studio*. 2022. URL: <https://visualstudio.microsoft.com/> (visited on 07/21/2022) (cit. on p. 1).
- [2] The Code::Blocks Team. *Code::Blocks*. 2020. URL: <http://www.codeblocks.org/> (visited on 06/05/2020) (cit. on p. 1).
- [3] Apple. *Xcode*. 2020. URL: <https://developer.apple.com/xcode/> (visited on 06/05/2020) (cit. on p. 1).
- [4] *Programming Language C: American National Standard for Information Systems : ANSI X3.159-1989*. A.N.S.I., 1989 (cit. on p. 1).

Index

`==`, operator, 6

A

Allman-stijl, 9

C

conditie, 5

D

debuggen, 1

E

else, keyword, 5

H

header-bestand, 4

I

if, keyword, 5

int, keyword, 3

K

keyword, 5

M

main, functie, 2, 4

R

recursie, i

return, keyword, 3

S

scanf, functie, 4

statement, 3

stdio.h, header-bestand, 4

V

variabele, 2

void, keyword, 2