**Opdracht 1.1**

|  |
| --- |
| void main(void)  {  short i = 20;  while(i>0){  printf("Hello World ");  i--;  }  \_getch();  } |

**Opdracht 1.2**

|  |
| --- |
| void main(void)  {  int a,b = 0;  printf("Voer A en B in als a,b: ");  scanf("%d,%d", &a, &b);  printf("\n%d", a+b);  \_getch();  } |

**Opdracht 1.3**

|  |
| --- |
| #define long unsigned long  void main(void){  long a = 1000;  long b = 1004;  printf("%d", a\*b);  \_getch();  } |

**Opdracht 1.4**

|  |
| --- |
| void main(void){  int num = 7;  short pow = 5;  long ref = num;  while(pow > 1){  ref \*= num;  pow--;  }  printf("%d", ref);  \_getch();  } |

**Opdracht 1.5 & 1.6**

|  |
| --- |
| void main(void){  int row[10] = {18, 72, 75, 25, 90, 15 ,27 , 70, 98, 28};  int cnt = 0;  short ref = 0;  while(ref<=9){  cnt +=row[ref];  ref++;  }  printf("Het eerste nummer is %d\nHet laatste nummer is %d\n", row[0], row[9]);  printf("Gemiddeld = %f", cnt/10.0);  \_getch();  } |

**Opdracht 1.7**

|  |
| --- |
| void main(void){  char voor[10], achter[10] = {'\0'};  printf("Voornaam: ");  scanf("%9s", voor);  printf("Achternaam: ");  scanf("%9s", achter);  printf("Welkom, %s %s", voor, achter);  \_getch();  } |

**Opdracht 1.8**

|  |
| --- |
| void main(void){  char seed = '\0';  printf("Seed char?");  scanf("%c", &seed);  while(true){  printf("%c ", seed);  seed++;  if(seed==91||seed==123)  break;  }  \_getch();  } |

**Opdracht 2.1**

|  |
| --- |
| void main(void){  int row[MAXROW] = {0};  for(int i = 0; i<MAXROW;i++){  printf("Getal: ");scanf("%d", &row[i]);  }  for(int i = MAXROW - 1;i>=0;i--){  printf("%d ", row[i]);  }  \_getch();  } |

**Opdracht 2.2**

|  |
| --- |
| void main(void){  int row[MAXROW] = {0};  for(int i = 0; i<MAXROW;i++){  printf("Getal: ");scanf("%d", &row[i]);  }  int high = 0,shigh = 0;  for(int i = 0;i<=MAXROW;i++){  int num = row[i];  if(num>high){  shigh = high;  high = num;  }  else if(num>shigh)  shigh=num;  }  printf("%d is het op een na hoogste", shigh);  \_getch();  } |

**Opdracht 2.3**

|  |
| --- |
| void main(void){  int a=0,b=0;  char num[3] ={'\0'};  printf("Getal, gescheiden door comma: ");  scanf("%d, %d", &a, &b);  itoa(a+b, num, 10);  if(a+b>=10)  {  /\* Switcharoo \*/  char middle = num[0];  num[0] = num[1];  num[1] = middle;  }  printf("%s", num);  \_getch();  } |

**Opdracht 2.4**

|  |
| --- |
| #define MAXROW 10  void main(void){  char str[50] = {'\0'};  printf("Lees zin in om te coderen");  scanf("%s", str);  short cnt = 0;  while(1)  {  char letter = str[cnt];  if(letter == '\0')  break;  if(letter<97)  letter+= 32;  int dif = letter - 97;  printf("%c", 122 - dif);  cnt++;  }  \_getch();  } |

**Opdracht 3.1**

**Main.h**

|  |
| --- |
| #include <iostream>  #include <stdlib.h>  #include <conio.h>  typedef struct Student {  char studentNummer[11];  char naam[20];  char opleiding[10];  short leeftijd;  Student\* next;  Student\* previous;  };  int menu(Student\*);  int enter(Student\*);  int read(Student\*);  int export(Student\*);  int import(Student\*);  void replacenewspaces(char\*); |

**Main.cpp**

|  |
| --- |
| #include "Main.h"  void filltest(Student\* studenten)  {  Student \*een = (Student\*)malloc(sizeof(Student));  {  strcpy(een->naam, "David Diks");  strcpy(een->opleiding, "E2ETC");  een->leeftijd = 18;  strcpy(een->studentNummer, "1233122");  een->previous = studenten;  studenten->next = een;  }  Student \*twee = (Student\*)malloc(sizeof(Student));  {  strcpy(twee->naam, "Henk Fassels");  strcpy(twee->opleiding, "E2E&A");  twee->leeftijd= 19;  strcpy(twee->studentNummer, "1325069");  twee->previous = een;  een->next = twee;  twee->next = NULL;  }  }  void main(void){  //Een student initialiseren  Student studenten = {};  FILE\* f = NULL;  f = fopen("records.txt", "ab+");  fclose(f);  while(1)  {  if(menu(&studenten) == '6')  break;  }  }  int menu(Student \*studenten)  {  fflush(stdin);  char keuze = 0;  printf("1: Inlezen\n2: Weergeven\n3: Exporteren\n4: Importeren\n5: Wissel\n6: Exit\n--------------------------\n: ");  scanf("%c", &keuze);  switch(keuze)  {  case '1':  if(enter(studenten) == 1)  printf("Student added\n");  break;  case '2':  read(studenten);  break;  case '3':  export(studenten);  break;  case '4':  import(studenten);  break;  case '6':  export(studenten);  break;  default:  break;  }  return keuze;  }  int enter(Student\* studenten)  {  //Laatste student vinden.  while(1)  {  if(studenten->next==NULL)  break;  else  studenten = studenten->next;  }  //Tijdelijke student om gegevens in te zetten  Student \*tijdelijk = (Student\*)malloc(sizeof(Student));  tijdelijk->previous = studenten; //Previous opzetten  tijdelijk->next = NULL;  fflush(stdin);  printf("Wat is de naam: ");  fgets(tijdelijk->naam, 19, stdin);  replacenewspaces(tijdelijk->naam);  printf("Opleiding: ");  fgets(tijdelijk->opleiding, 9, stdin);  replacenewspaces(tijdelijk->opleiding);  printf("Studentennummer: ");  fgets(tijdelijk->studentNummer, 10, stdin);  replacenewspaces(tijdelijk->studentNummer);  printf("Wat is de leeftijd: ");  scanf("%d", &tijdelijk->leeftijd);  fflush(stdin);  char cor = 'n';  printf("Is dit corrent (Y/n): ");  scanf("%c", &cor);  if(cor != 'n')  studenten->next = tijdelijk; //Laatste element in het echte archief naar nieuw blokje leiden. Als het goed is wordt deze dan niet door GC opgehaald  else  return 0;  return 1;  }  int read(Student \*studenten)  {  while(1)  {  if(studenten->previous != NULL)  {  fflush(stdin);  char chc = 'q';  printf("--------------------------------\nNaam: %19s\nOpleiding: %9s\nStudentennummer: %10s\nLeeftijd: %d\n", studenten->naam, studenten->opleiding, studenten->studentNummer, studenten->leeftijd);  printf("[v]olgende of [q]uitten: ");  chc = getchar();  if(chc == 'q')  break;  }  //Kijken of de volgende student bestaat  if(studenten->next == NULL)  break;  studenten = studenten ->next;  }  return 1;  }  int export(Student\* studenten)  {  FILE\* f = NULL;  f = fopen("records.txt", "w");  if(f != NULL)  {  while(1)  {  if(studenten->previous != NULL)  {  char leeftijd[3] = {'\n'};  itoa(studenten->leeftijd, leeftijd, 10);  fwrite(studenten->naam,sizeof(char), sizeof(studenten->naam), f);  fwrite(studenten->opleiding, sizeof(char), sizeof(studenten->opleiding), f);  fwrite(leeftijd, sizeof(char), sizeof(leeftijd), f);  fwrite(studenten->studentNummer, sizeof(char), sizeof(studenten->studentNummer), f);  fwrite("\n", 1, 1, f);  }  if(studenten->next == NULL)  break;  studenten = studenten->next;  }  }  fclose(f);  return 1;  }  int import(Student \*studenten)  {  while(1)  {  if(studenten->next == NULL)  break;  else  studenten = studenten->next;  }  FILE \*f = NULL;  f = fopen("records.txt", "r");  while(1)  {  char buffer[50] = {'\0'};  char leeftijdBuffer[3] = {'\0'};  fgets(buffer, sizeof(buffer), f);  if(buffer[0] == '\0')  break;  //Naam uitlezen  Student \*tijdelijk = (Student\*)malloc(sizeof(Student));  strcpy(tijdelijk->naam, &buffer[0]);  strcpy(tijdelijk->opleiding, &buffer[20]);  strcpy(leeftijdBuffer, &buffer[30]);  strcpy(tijdelijk->studentNummer, &buffer[33]);  tijdelijk->leeftijd = atoi(leeftijdBuffer);  studenten->next = tijdelijk;  tijdelijk->previous = studenten;  studenten = tijdelijk;  studenten->next = NULL;  }  fclose(f);  return 1;  }  void replacenewspaces(char \*name)  {  size\_t ln = strlen(name) - 1;  if (name[ln] == '\n')  name[ln] = '\0';  } |