

24 Februari 2024

PRAKTIKUM 1.1 dan 1.2

Mata Kuliah Algoritma dan Struktur Data



Nama Dosen Pengampu

Ibu Umi Sa'adah S.Kom, M.Kom

Dikerjakan oleh

Nama : Muhammad Rafi Rizaldi

NRP : 3123600001

Kelas : 1 D4 IT A

PRAKTIKUM 1.1

1. Dengan menggunakan array of struct, buatlah program sbb:

Input data-data mahasiswa yang terdiri atas :

nama, nilai Tugas, nilai UTS, nilai UAS sebanyak n mahasiswa

Output :

Tampilkan dalam bentuk tabel

No, NRP, Nama Mhs, Nilai tugas, nilai UTS, nilai UAS, nilai Akhir, Grade

(Nilai Akhir = 20% tugas + 40% UTS + 40%UAS) dengan ketentuan sbb :

- Nilai akhir ≥ 80 Grade A

- Nilai akhir ≥ 70 Grade B

- Nilai akhir ≥ 60 Grade C

- Nilai akhir ≥ 50 Grade D

- Nilai akhir < 50 Grade E

Petunjuk :

- Gunakan array of struct (tanpa typedef)

- Perhatikan contoh tampilan

- Nilai tambah, gunakan array sebagai parameter fungsi (misal 2 fungsi : input() dan tampil())

Source code :

```
#include <stdlib.h>
#include <string.h>
#include <stdio.h>

struct Data
{
    char nama[50];
    float nilai_tugas;
    float nilai_uts;
    float nilai_uas;
    float nilai_akhir;
    char grade;
```

```

};

void PrintLine();
void InputData(int *, struct Data **);
char GetGrade(float);
void PrintResult(int, struct Data *);
void clearScreen();

int main(int argc, char const *argv[])
{
    int jumlah_mahasiswa;
    struct Data *mahasiswa;
    InputData(&jumlah_mahasiswa, &mahasiswa);
    PrintResult(jumlah_mahasiswa, mahasiswa);

    free(mahasiswa);
    mahasiswa = NULL;
    return 0;
}

void InputData(int *jumlah, struct Data **data)
{
    printf("MENGHITUNG NILAI AKHIR \nMATA KULIAH KONSEP\nPEMROGRAMAN\n\n");
    printf("Berapa jumlah mahasiswa ? ");
    scanf("%d", jumlah);
    getchar();

    *data = (struct Data *)malloc(*jumlah * sizeof(struct Data));

    if (*data == NULL)
        printf("gagal mengalokasikan memori\n");

    printf("MASUKKAN DATA MAHASISWA\n\n");
    for (int i = 0; i < *jumlah; i++)
    {
        printf("Mahasiswa ke - %d\n", i + 1);
        printf("Nama : ");
        fgets((*data)[i].nama, sizeof((*data)[i].nama), stdin);
        (*data)[i].nama[strcspn((*data)[i].nama, "\n")] = 0;
        printf("Nilai Tugas : ");
        scanf("%f", &(*data)[i].nilai_tugas);
        printf("Nilai UTS : ");
    }
}

```

```

        scanf("%f", &(*data)[i].nilai_uts);
        printf("Nilai UAS : ");
        scanf("%f", &(*data)[i].nilai_uas);
        getchar();
        (*data)[i].nilai_akhir = (*data)[i].nilai_tugas * 0.2 +
(*data)[i].nilai_uts * 0.4 + (*data)[i].nilai_uas * 0.4;
        (*data)[i].grade = GetGrade((*data)[i].nilai_akhir);
    }
    clearScreen();
}

```

```

void PrintResult(int jumlah, struct Data *data)

```

```

{
    printf("\t\t\t\tDAFTAR NILAI\n");
    printf("\t\t\tMATAKULIAH KONSEP PEMROGRAMAN\n");
    PrintLine();

    printf("No\tNama\t\t\tNilai\t\t\tGrade\n");
    printf("\tMahasiswa\tTugas\tUTS\tUAS\tAkhir\n");
    PrintLine();

    for (int i = 0; i < jumlah; i++)
    {
        printf("%d\t%s\t\t%.1f\t%.1f\t%.1f\t%.1f\t%c\n", i + 1,
data[i].nama, data[i].nilai_tugas, data[i].nilai_uts,
data[i].nilai_uas, data[i].nilai_akhir, data[i].grade);
    }
}

```

```

char GetGrade(float akhir)

```

```

{
    if (akhir >= 80)
        return 'A';
    else if (akhir >= 70 && akhir < 80)
        return 'B';
    else if (akhir >= 60 && akhir < 70)
        return 'C';
    else if (akhir >= 50 && akhir < 60)
        return 'D';
    else
        return 'E';
}

```

```

void PrintLine()
{
    for (int i = 0; i < 80; i++)
    {
        printf("-");
    }
    printf("\n");
}

void clearScreen()
{
#ifdef _WIN32
    system("cls"); // For Windows
#else
    printf("\033[2J\033[H"); // For Linux/Unix
#endif
}

```

Output :

```

o eros@Rafi-Rizaldi:~/KULIAH2/ASD/prakt1.1$ ./tes
MENGHITUNG NILAI AKHIR
MATA KULIAH KONSEP PEMROGRAMAN

Berapa jumlah mahasiswa ? 3
MASUKKAN DATA MAHASISWA

Mahasiswa ke - 1
Nama : anas
Nilai Tugas : 76
Nilai UTS : 86
Nilai UAS : 56
Mahasiswa ke - 2
Nama : wawa
Nilai Tugas : 85
Nilai UTS : 78
Nilai UAS : 99
Mahasiswa ke - 3
Nama : satrio
Nilai Tugas : 76
Nilai UTS : 65
Nilai UAS : 55

```

DAFTAR NILAI MATAKULIAH KONSEP PEMROGRAMAN						
No	Nama Mahasiswa	Nilai				Grade
		Tugas	UTS	UAS	Akhir	
1	anas	76.0	86.0	56.0	72.0	B
2	wawa	85.0	78.0	99.0	87.8	A
3	satrio	76.0	65.0	55.0	63.2	C

2. Dengan menggunakan typedef, buatlah tipe data struct untuk menyimpan informasi pegawai (misal tipe pegawai) berupa : no ID, nama, tgl lahir, jenis kelamin (L/P) dan gaji/bulan.

- di main() deklarasikan sebuah var bertipe array of pegawai (misal emp[])
- panggil fungsi input() untuk memasukkan data-datanya dengan pengiriman parameter secara pass by reference
- panggil fungsi tampil() untuk menampilkan semua data yang telah diinputkan dengan pengiriman parameter secara pass by reference

Petunjuk :

- Untuk no id, gunakan auto increment (tanpa input) perhatikan contoh tampilan
- Jika diperlukan, gunakan fungsi strcpy() dari <string.h>

Source code :

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

typedef struct {
    int dd, mm, yyyy;
}Date;

typedef struct info
{
    int id;
    char nama[40];
    Date ttl;
    char gender;
    int gaji;
} Informasi;
```

```

void InputData(Informasi **, int *);
void PrintResult(Informasi *, int);
void clearScreen();

int main(int argc, char const *argv[])
{
    printf("DATA PEGAWAI\n\n");
    int jumlah_pegawai;
    Informasi *pegawai;

    InputData(&pegawai, &jumlah_pegawai);
    PrintResult(pegawai, jumlah_pegawai);
    return 0;
}

void InputData(Informasi **data, int *jumlah)
{
    printf("Berapa jumlah pegawai : ");
    scanf("%d", jumlah);
    getchar();

    *data = (Informasi *)malloc(*jumlah * (sizeof(Informasi)));
    if (*data == NULL)
    {
        printf("gagal alokasimemori\n");
    }

    for (int i = 0; i < *jumlah; i++)
    {
        printf("Data pegawai ke-%d\n", i + 1);
        printf("Nama \t\t\t: ");
        fgets((*data)[i].nama, sizeof((*data)[i].nama), stdin);
        (*data)[i].nama[strcspn((*data)[i].nama, "\n")] = 0;
        printf("Tgl lahir (dd-mm-yy)\t: ");
        scanf("%d-%d-%d", &(*data)[i].ttl.dd, &(*data)[i].ttl.mm,
&(*data)[i].ttl.yyyy);
        getchar();
        printf("Jenis kelamin (L/P)\t: ");
        scanf("%c", &(*data)[i].gender);
        printf("Gaji/bulan \t\t: ");
        scanf("%d", &(*data)[i].gaji);
        getchar();
    }
}

```

```

}
// clearScreen();
}

void PrintResult(Informasi *data, int jumlah)
{
    printf("Data pegawai yang diinputkan \n\n");
    for (int i = 0; i < jumlah; i++)
    {
        printf("No ID \t\t: %d\n", i + 1);
        printf("Nama \t\t: %s\n", data[i].nama);
        printf("Tgl lahir \t: %d-%d-%d\n", data[i].ttl.dd, data[i].ttl.mm,
data[i].ttl.yyyy);
        printf("Jenis Kelamin\t: %s\n", (data[i].gender == 'l' ||
data[i].gender == 'L') ? "Laki-laki " : "Perempuan");
        printf("Gaji/bulan \t: Rp.%d\n\n", data[i].gaji);
    }
}

void clearScreen()
{
#ifdef _WIN32
    system("cls"); // For Windows
#else
    printf("\033[2J\033[H"); // For Linux/Unix
#endif
}

```

Output :

DATA PEGAWAI

Berapa jumlah pegawai : 3

Data pegawai ke-1

Nama : anas
Tgl lahir (dd-mm-yy) : 22-5-2000
Jenis kelamin (L/P) : l
Gaji/bulan : 1000000

Data pegawai ke-2

Nama : tata
Tgl lahir (dd-mm-yy) : 23-5-1999
Jenis kelamin (L/P) : p
Gaji/bulan : 2000000

Data pegawai ke-3

Nama : masha
Tgl lahir (dd-mm-yy) : 14-2-2001
Jenis kelamin (L/P) : p
Gaji/bulan : 1500000

Data pegawai yang diinputkan

No ID : 1
Nama : anas
Tgl lahir : 22-5-2000
Jenis Kelamin : Laki-laki
Gaji/bulan : Rp.1000000

No ID : 2
Nama : tata
Tgl lahir : 23-5-1999
Jenis Kelamin : Perempuan
Gaji/bulan : Rp.2000000

No ID : 3
Nama : masha
Tgl lahir : 14-2-2001
Jenis Kelamin : Perempuan
Gaji/bulan : Rp.1500000

3. Modifikasi program dari soal no 1.

- Ubah nama tipe data dengan menggunakan typedef
- Panggillah fungsi input() dan tampil() dengan cara pass by reference

Petunjuk :

- Untuk soal nomor 2 & 3, perhatikan cara membaca array dengan menggunakan pointer, buka kembali modul teori pointer to array

Source code :

```
#include <stdlib.h>
#include <string.h>
#include <stdio.h>

typedef struct data
{
    char nama[50];
    float nilai_tugas;
    float nilai_uts;
    float nilai_uas;
    float nilai_akhir;
    char grade;
} Data;

void PrintLine();
void InputData(int *, Data **);
char GetGrade(float);
void PrintResult(int, Data *);
void clearScreen();

int main(int argc, char const *argv[])
{
    int jumlah_mahasiswa;
    Data *mahasiswa;
    InputData(&jumlah_mahasiswa, &mahasiswa);
    PrintResult(jumlah_mahasiswa, mahasiswa);

    free(mahasiswa);
    mahasiswa = NULL;
    return 0;
}
```

```

void InputData(int *jumlah, Data **data)
{
    printf("MENGHITUNG NILAI AKHIR \nMATA KULIAH KONSEP PEMROGRAMAN\n\n");
    printf("Berapa jumlah mahasiswa ? ");
    scanf("%d", jumlah);
    getchar();

    *data = (Data *)malloc(*jumlah * sizeof(Data));

    if (*data == NULL)
        printf("gagal mengalokasikan memori\n");

    printf("MASUKKAN DATA MAHASISWA\n\n");
    for (int i = 0; i < *jumlah; i++)
    {
        printf("Mahasiswa ke - %d\n", i + 1);
        printf("Nama : ");
        fgets((*data)[i].nama, sizeof((*data)[i].nama), stdin);
        (*data)[i].nama[strcspn((*data)[i].nama, "\n")] = 0;
        printf("Nilai Tugas : ");
        scanf("%f", &(*data)[i].nilai_tugas);
        printf("Nilai UTS : ");
        scanf("%f", &(*data)[i].nilai_uts);
        printf("Nilai UAS : ");
        scanf("%f", &(*data)[i].nilai_uas);
        getchar();
        (*data)[i].nilai_akhir = (*data)[i].nilai_tugas * 0.2 +
        (*data)[i].nilai_uts * 0.4 + (*data)[i].nilai_uas * 0.4;
        (*data)[i].grade = GetGrade((*data)[i].nilai_akhir);
    }
    clearScreen();
}

void PrintResult(int jumlah, Data *data)
{
    printf("\t\t\t\tDAFTAR NILAI\n");
    printf("\t\t\t\tMATAKULIAH KONSEP PEMROGRAMAN\n");
    PrintLine();

    printf("No\tNama\t\t\t\tNilai\t\t\t\tGrade\n");
    printf("\tMahasiswa\tTugas\tUTS\tUAS\tAkhir\n");
    PrintLine();
}

```

```

for (int i = 0; i < jumlah; i++)
{
    printf("%d\t%s\t\t%.1f\t%.1f\t%.1f\t%.1f\t%c\n", i + 1,
data[i].nama, data[i].nilai_tugas, data[i].nilai_uts,
data[i].nilai_uas, data[i].nilai_akhir, data[i].grade);
}

char GetGrade(float akhir)
{
    if (akhir >= 80)
        return 'A';
    else if (akhir >= 70 && akhir < 80)
        return 'B';
    else if (akhir >= 60 && akhir < 70)
        return 'C';
    else if (akhir >= 50 && akhir < 60)
        return 'D';
    else
        return 'E';
}

void PrintLine()
{
    for (int i = 0; i < 80; i++)
    {
        printf("-");
    }
    printf("\n");
}

void clearScreen()
{
#ifdef _WIN32
    system("cls"); // For Windows
#else
    printf("\033[2J\033[H"); // For Linux/Unix
#endif
}

```

Output :

```
eros@Rafi-Rizaldi:~/KULIAH2/ASD/prakt1.1$ ./tes
MENGHITUNG NILAI AKHIR
MATA KULIAH KONSEP PEMROGRAMAN

Berapa jumlah mahasiswa ? 3
MASUKKAN DATA MAHASISWA

Mahasiswa ke - 1
Nama : anas
Nilai Tugas : 76
Nilai UTS : 86
Nilai UAS : 56
Mahasiswa ke - 2
Nama : wawa
Nilai Tugas : 85
Nilai UTS : 78
Nilai UAS : 99
Mahasiswa ke - 3
Nama : satrio
Nilai Tugas : 76
Nilai UTS : 65
Nilai UAS : 55
```

DAFTAR NILAI MATAKULIAH KONSEP PEMROGRAMAN						
No	Nama Mahasiswa	Nilai			Grade	
		Tugas	UTS	UAS		
1	anas	76.0	86.0	56.0	72.0	B
2	wawa	85.0	78.0	99.0	87.8	A
3	satrio	76.0	65.0	55.0	63.2	C

4. Untuk soal no 3 diatas, pada tampilan output, tambahkan informasi : nama & nilai akhir mahasiswa yang tertinggi (lihat contoh)

Petunjuk:

- buatlah tipe data struct baru misal dengan nama maks dengan 2 field : nama & nilai untuk menyimpan informasi mahasiswa dengan nilai tertinggi
- Sertakan <string.h> agar bisa menggunakan fungsi strcpy() untuk mengcopy nama

Source Code :

```

#include <stdlib.h>
#include <string.h>
#include <stdio.h>

typedef struct data
{
    char nama[50];
    float nilai_tugas;
    float nilai_uts;
    float nilai_uas;
    float nilai_akhir;
    char grade;
} Data;

typedef struct
{
    char nama[50];
    float nilai_akhir;
} Maks;

void PrintLine();
void InputData(int *, Data **);
char GetGrade(float);
void PrintResult(int, Data *);
void FindTheBest(Data *, int);

int main(int argc, char const *argv[])
{
    int jumlah_mahasiswa;
    Data *mahasiswa;
    InputData(&jumlah_mahasiswa, &mahasiswa);
    PrintResult(jumlah_mahasiswa, mahasiswa);
    FindTheBest(mahasiswa, jumlah_mahasiswa);

    free(mahasiswa);
    mahasiswa = NULL;
    return 0;
}

void InputData(int *jumlah, Data **data)
{
    printf("MENGHITUNG NILAI AKHIR \nMATA KULIAH KONSEP PEMROGRAMAN\n\n");

```

```

printf("Berapa jumlah mahasiswa ? ");
scanf("%d", &jumlah);
getchar();

*data = (Data *)malloc(*jumlah * sizeof(Data));

if (*data == NULL)
    printf("gagal mengalokasikan memori\n");

printf("MASUKKAN DATA MAHASISWA\n\n");
for (int i = 0; i < *jumlah; i++)
{
    printf("Mahasiswa ke - %d\n", i + 1);
    printf("Nama : ");
    fgets((*data)[i].nama, sizeof((*data)[i].nama), stdin);
    (*data)[i].nama[strcspn((*data)[i].nama, "\n")] = 0;
    printf("Nilai Tugas : ");
    scanf("%f", &(*data)[i].nilai_tugas);
    printf("Nilai UTS : ");
    scanf("%f", &(*data)[i].nilai_uts);
    printf("Nilai UAS : ");
    scanf("%f", &(*data)[i].nilai_uas);
    getchar();
    (*data)[i].nilai_akhir = (*data)[i].nilai_tugas * 0.2 +
    (*data)[i].nilai_uts * 0.4 + (*data)[i].nilai_uas * 0.4;
    (*data)[i].grade = GetGrade((*data)[i].nilai_akhir);
}
}

void PrintResult(int jumlah, Data *data)
{
    printf("\t\t\t\tDAFTAR NILAI\n");
    printf("\t\t\t\tMATAKULIAH KONSEP PEMROGRAMAN\n");
    PrintLine();

    printf("No\tNama\t\t\t\tNilai\t\t\t\tGrade\n");
    printf("\tMahasiswa\tTugas\tUTS\tUAS\tAkhir\n");
    PrintLine();

    for (int i = 0; i < jumlah; i++)
    {

```

```

        printf("%d\t%s\t\t%.1f\t%.1f\t%.1f\t%.1f\t%c\n", i + 1,
data[i].nama, data[i].nilai_tugas, data[i].nilai_uts,
data[i].nilai_uas, data[i].nilai_akhir, data[i].grade);
    }
}

```

```

char GetGrade(float akhir)

```

```

{
    if (akhir >= 80)
        return 'A';
    else if (akhir >= 70 && akhir < 80)
        return 'B';
    else if (akhir >= 60 && akhir < 70)
        return 'C';
    else if (akhir >= 50 && akhir < 60)
        return 'D';
    else
        return 'E';
}

```

```

void FindTheBest(Data *data, int total_data)

```

```

{
    Maks hasil;
    float maks = __FLT_MIN__;
    int posisi;
    for (int i = 0; i < total_data; i++)
    {
        if (data[i].nilai_akhir > maks)
        {
            maks = data[i].nilai_akhir;
            posisi = i;
        }
    }
    hasil.nilai_akhir = maks;
    strcpy(hasil.nama, data[posisi].nama);
    printf("\n\nTotal mahasiswa = %d\n\n", total_data);
    printf("Nilai tertinggi\n");
    printf("Nama \t= %s\n", hasil.nama);
    printf("Nilai \t= %.1f\n", hasil.nilai_akhir);
}

```

```

void PrintLine()

```

```

{
    for (int i = 0; i < 80; i++)

```



```

{
    printf("-");
}
printf("\n");
}

```

Output :

```

MENGHITUNG NILAI AKHIR
MATA KULIAH KONSEP PEMROGRAMAN

```

```

Berapa jumlah mahasiswa ? 3
MASUKKAN DATA MAHASISWA

```

```

Mahasiswa ke - 1
Nama : AKMAL
Nilai Tugas : 77
Nilai UTS : 88
Nilai UAS : 99
Mahasiswa ke - 2
Nama : GERUNG
Nilai Tugas : 88
Nilai UTS : 77
Nilai UAS : 56
Mahasiswa ke - 3
Nama : JERO
Nilai Tugas : 99
Nilai UTS : 77
Nilai UAS : 88

```

```

          DAFTAR NILAI
MATAKULIAH KONSEP PEMROGRAMAN

```

No	Nama Mahasiswa	Tugas	Nilai UTS	UAS	Akhir	Grade
1	AKMAL	77.0	88.0	99.0	90.2	A
2	GERUNG	88.0	77.0	56.0	70.8	B
3	JERO	99.0	77.0	88.0	85.8	A

```

Total mahasiswa = 3

```

```

Nilai tertinggi
Nama    = AKMAL
Nilai   = 90.2

```

PRAKTIKUM 1.2

1. Menggunakan fungsi sizeof() untuk menentukan ukuran objek

```
#include <stdio.h>
typedef struct employee_st
{
    char name[40];
    int id;
} Employee;
int main()
{
    int myInt;
    Employee john;
    printf("Size of int is %d\n", sizeof(myInt));
    // The argument of sizeof is an object
    printf("Size of int is %d\n", sizeof(int));
    // The argument of sizeof is a data type
    printf("Size of Employee is %d\n", sizeof(Employee));
    // The argument of sizeof is an object
    printf("Size of john is %d\n", sizeof(john));
    // The argument of sizeof is a data type
    printf("Size of char is %d\n", sizeof(char));
    printf("Size of short is %d\n", sizeof(short));
    printf("Size of int is %d\n", sizeof(int));
    printf("Size of long is %d\n", sizeof(long));
    printf("Size of float is %d\n", sizeof(float));
    printf("Size of double is %d\n", sizeof(double));
}
```

Output :

```

● mrrizzzz@Draverezld:/mnt/d/KULIAH2-ASD/ASD/prakt1.2$ ./tes
Size of int is 4
Size of int is 4
Size of Employee is 44
Size of john is 44
Size of char is 1
Size of short is 2
Size of int is 4
Size of long is 8
Size of float is 4
Size of double is 8

```

2. Menggunakan fungsi malloc() untuk mengalokasikan memory

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main()
{
    char s1[] = "This is a sentence";
    char *pblok;
    pblok = (char *)malloc(strlen((s1) + 1));

    if (pblok == NULL)
        printf("Error on malloc\n");
    else
    {
        strcpy(pblok, s1);
        printf("s1: %s\n", s1);
        printf("pblok: %s\n", pblok);
    }
}

```

Output :

```

● mrrizzzz@Draverezld:/mnt/d/K
s1: This is a sentence
pblok: This is a sentence

```

3. Menggunakan fungsi free() untuk membebaskan kembali memory

```

#include <stdio.h>

```

```

#include <stdlib.h>
int main()
{
    char *pblok;
    pblok = (char *)malloc(500 * sizeof(char));
    if (pblok == NULL)
        puts("Error on malloc");
    else
    {
        puts("OK, alokasi memory sudah dilakukan");
        puts("-----");
        free(pblok);
        pblok = NULL;
        puts("Blok memory telah dibebaskan kembali");
        puts("dan pointernya sdh digroundkan");
    }
}

```

Output :

```

● mrrizzz@Draverezld:/mnt/d/KULIAH2-ASD/ASD
OK, alokasi memory sudah dilakukan
-----
Blok memory telah dibebaskan kembali
dan pointernya sdh digroundkan

```

4. Mengalokasikan memory untuk objek struct dengan jumlah dan data yang diinputkan kemudian menampilkannya

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
typedef struct employee_st
{
    char name[40];
    int id;
} Employee;
int main()
{
    Employee *workers, *wpt;
    int num, i;

```

```

printf("How many employees do you want ? ");
scanf("%d", &num);
// getchar();
workers = (Employee *)malloc(num * sizeof(Employee));
if (workers == NULL)
{
    printf("Unable to allocated space for employees\n");

    exit(0);
}
wpt = workers;
for (i = 1; i <= num; i++)
{
    fflush(stdin);
    getchar();
    printf("Employee's name : ");
    gets(wpt->name);
    printf("Employee's id : ");
    scanf("%d", &wpt->id);
    wpt++;
}
puts("");
wpt = workers;
for (i = 1; i <= num; i++)
{
    printf("Employee %d is %s\n", wpt->id, wpt->name);
    wpt++;
}
free(workers);
workers = NULL;
}

```

Output :

```

● mrrizzz@Draverez1d:/mnt/d/KULIAH2-ASD/A
How many employees do you want ? 3
Employee's name : roger
Employee's id : 3
Employee's name : miya
Employee's id : 2
Employee's name : lolita
Employee's id : 1

Employee 3 is roger
Employee 2 is miya
Employee 1 is lolita

```

q4.

Source code :

```

#include <stdio.h>

typedef struct
{
    int day, month, year;
} Date;

int sameDay(Date, Date);

int main(int argc, char const *argv[])
{
    Date first_date, second_date;
    printf("Masukkan tanggal pertama (dd-mm-yyyy): ");
    scanf("%d-%d-%d", &first_date.day, &first_date.month,
    &first_date.year);
    printf("Masukkan tanggal kedua (dd-mm-yyyy): ");
    scanf("%d-%d-%d", &second_date.day, &second_date.month,
    &second_date.year);
    printf(sameDay(first_date, second_date) == 1 ? "It is the same day\n"
: "It is not the same day\n");

    return 0;
}

int sameDay(Date first_date, Date second_date)
{

```

```

    return (first_date.day == second_date.day && first_date.month ==
second_date.month && first_date.year == second_date.year) ? 1 : 0;
}

```

Output :

```

● mrrizzz@Draverezld:/mnt/d/KULIAH2-ASD/ASD/prakt1.2$ ./tes
Masukkan tanggal pertama (dd-mm-yyyy): 22-4-2000
Masukkan tanggal kedua (dd-mm-yyyy): 22-4-2001
It is not the same day
● mrrizzz@Draverezld:/mnt/d/KULIAH2-ASD/ASD/prakt1.2$ ./tes
Masukkan tanggal pertama (dd-mm-yyyy): 22-4-2000
Masukkan tanggal kedua (dd-mm-yyyy): 22-4-2000
It is the same day

```

q5.

Source code :

```

#include <stdio.h>

typedef struct
{
    int day, month, year;
} Date;

int isYounger(Date, Date);

int main(int argc, char const *argv[])
{
    Date first_student, second_student;
    printf("Input the first student's birth date (dd-mm-yyyy): ");
    scanf("%d-%d-%d", &first_student.day, &first_student.month,
&first_student.year);
    printf("Input the second student's birth date (dd-mm-yyyy): ");
    scanf("%d-%d-%d", &second_student.day, &second_student.month,
&second_student.year);
    printf(isYounger(first_student, second_student) == 1 ? "Yes, student
1 is younger than student 2\n" : "No, student 1 is not younger than
student 2\n");

    return 0;
}

```

```

}
int isYounger(Date student1, Date student2)
{
    if (student1.year < student2.year)
        return 1;
    else if (student1.year == student2.year)
    {
        if (student1.month < student2.month)
            return 1;
        else if (student1.month == student2.month)
        {
            if (student1.day < student2.day)
                return 1;
            else
                return 0;
        }
        else
            return 0;
    }
    else
        return 0;
}

```

Output :

```

● mrrizzza@Draverez1d:/mnt/d/KULIAH2-ASD/ASD/prakt1.2$ ./tes
Input the first student's birth date (dd-mm-yyyy): 12-4-2000
Input the second student's birth date (dd-mm-yyyy): 12-3-2001
Yes, student 1 is younger than student 2

```

q6.



