**Laporan Resmi**

**Praktikum Algoritma dan Struktur Data**

**Memory Allocation** 

**Dr. Tita Karlita S.Kom, M.Kom**

**Nama : Marits Ikmal Yasin**

**Kelas : 1D4 IT B**

**NRP : 3121600047**

1. Soal Nomor 1

Code :

#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*));

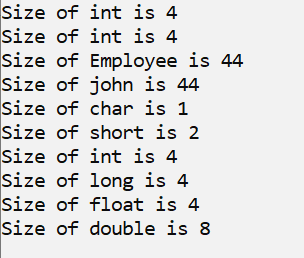
    printf("Size of long is *%d*\n", **sizeof**(*long*));

    printf("Size of float is *%d*\n", **sizeof**(*float*));

    printf("Size of double is *%d*\n", **sizeof**(*double*));

}

Output :



1. Soal Nomor 2

Code :

#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);

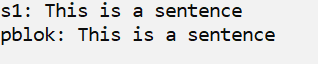
        printf("s1: *%s*\n", s1);

        printf("pblok: *%s*\n", pblok);

    }

}

Output :



1. Soal Nomor 3

Code :

#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;

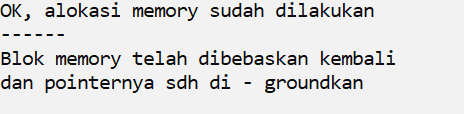
        puts("Blok memory telah dibebaskan kembali");

        puts("dan pointernya sdh di - groundkan");

    }

}

Output :



1. Soal Nomor 4

Code :

#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);

    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);

        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**++**;

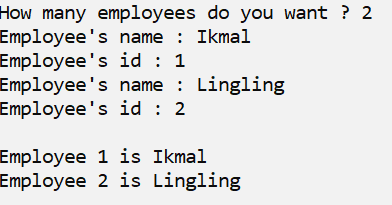
    }

    free(workers);

    workers **=** NULL;

}

Output :



1. Soal Nomor 5

Code :

#include<stdio.h>

typedef struct{

*int* tgl, bln, thn;

}date;

*int* sameday(date, date);

*int* main(){

    date day1, day2;

    printf("Masukkan Tanggal pertama dd-mm-yyyy: ");

    scanf("*%d*-*%d*-*%d*", **&**day1.*tgl*, **&**day1.*bln*, **&**day1.*thn*);

    printf("Masukkan Tanggal Kedua dd-mm-yyyy: ");

    scanf("*%d*-*%d*-*%d*", **&**day2.*tgl*, **&**day2.*bln*, **&**day2.*thn*);

**if**(sameday(day1, day2))

        printf("It is the same day\n");

**else**

        printf("It is not the same day\n");

**return** 0;

}

*int* sameday(date **hari1**, date **hari2**){

**if**(hari1.*tgl* **==** hari2.*tgl* **&&** hari1.*bln* **==** hari1.*bln* **&&** hari1.*thn* **==** hari2.*thn*)

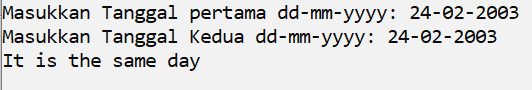
**return** 1;

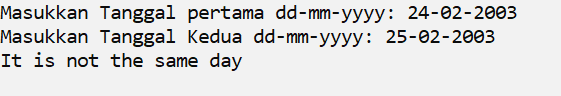
**else**

**return** 0;

};

Output :





1. Soal Nomor 6

Code :

#include<stdio.h>

typedef struct{

*int* tgl, bln, thn;

}birthday;

*int* isYounger(birthday, birthday);

*int* main(){

    birthday student1, student2;

    printf("Masukkan Tanggal Lahir Student 1 dd-mm-yyyy: ");

    scanf("*%d*-*%d*-*%d*", **&**student1.*tgl*, **&**student1.*bln*, **&**student1.*thn*);

    printf("Masukkan Tanggal Lahir Student 2 dd-mm-yyyy: ");

    scanf("*%d*-*%d*-*%d*", **&**student2.*tgl*, **&**student2.*bln*, **&**student2.*thn*);

**if**(isYounger(student1, student2))

        printf("Student 1 is younger than student 2\n");

**else**

        printf("Student 1 isn't younger than student 2\n");

**return** 0;

}

*int* isYounger(birthday **stud1**, birthday **stud2**){

**if**(stud1.*thn* **>** stud2.*thn*)

**return** 1;

**else** **if**(stud1.*thn* **==** stud2.*thn*){

**if**(stud1.*bln* **>** stud2.*bln*)

**return** 1;

**else** **if**(stud1.*bln* **==** stud2.*bln*){

**if**(stud1.*tgl* **>** stud2.*tgl*)

**return** 1;

**else**

**return** 0;

        }

**else**

**return** 0;

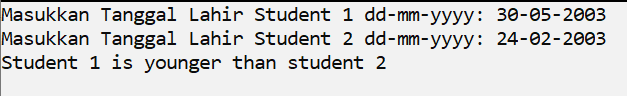
    }

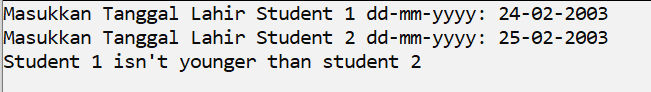
**else**

**return** 0;

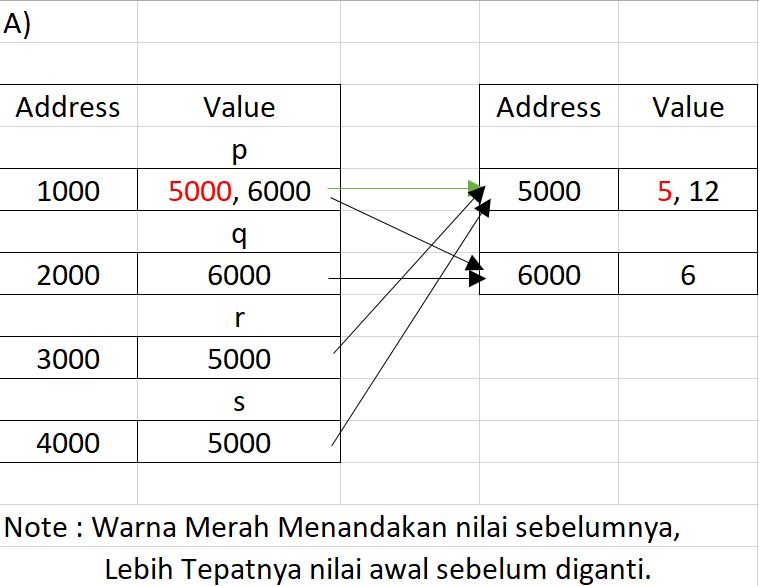
};

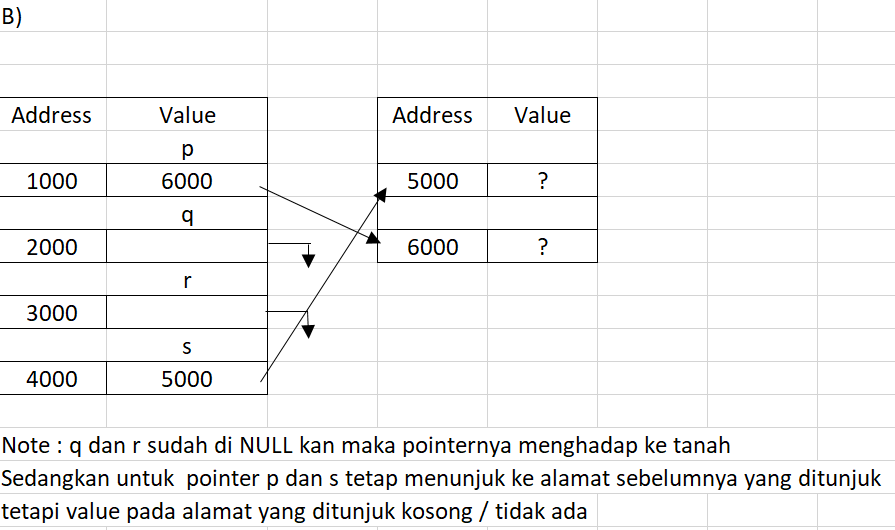
Output :





1. Soal Nomor 7





Analisis :

Dari praktikum ini, kita mempelajari bagaimana alokasi memori apabila menggunakan fungsi malloc pada bahasa C. Fungsi malloc itu sendiri yaitu berfungsi untuk memesan sebuah alokasi memori sebesar size yang kita inginkan. Size itu tergantung sesuai kebutuhan programmer. Bisa menggunakan sizeof dari tipe datanya maupun bisa langsung angka yang kita inginkan. Sizeof sendiri memudahkan kita untuk mengurangi terjadinya kegagalan pemesanan karena kita tidak perlu menghitung satu-persatu size yang kita inginkan.