Disusun oleh : F_Rofiqy

Program dapat diakses pada :

Question 1

Not yet answered

Marked out of 1.0

Flag question

Berikut adalah potongan program untuk membaca data dan memasukkan ke dalam array 2 dimensi berukuran N x M:

```
for (int i = 0; i < N; i++)
for (int j = 0; j < M; j++)
{
        scanf("%d", &nilai);
        arr[i][j] = nilai;
}
```

Untuk itu, deklarasi dan definisi variabel array tersebut yang salah adalah .....

```
int **arr = (int **)malloc(r*sizeof(int *));
for (i=0; i<r; i++)
    arr[i] = (int *)malloc(c*sizeof(int));
```

```
int arr[N][M]={5};
```

```
int *arr[N];
for (int i=0; i<N; i++)
    arr[i] = (int *)malloc(M*sizeof(int));
```

```
int arr[N][M];
```

```
arr = (int *)malloc(N * M * sizeof(int));
```

Question 2

Not yet answered

Marked out of 1.0

Flag question

Return type dari fungsi malloc() adalah .....

- ✗ `void *`
- `void **`
- `int *`
- `int **`
- `void`

No need to cast return value of `malloc` as its return type is `void*`.

33

> Can someone explain why do some programmers use `(char *)` in front of the malloc?

They are doing wrong (most probably) by casting it (in good programmers opinion).

As **wiki** says:

`malloc` returns a void pointer (`void *`), which indicates that it is a pointer to a region of unknown data type. **The use of casting is required in C++ due to the strong type system, whereas this is not the case in C**[1]. The lack of a specific pointer type returned from `malloc` is type-unsafe behavior according to some programmers: `malloc` allocates based on byte count but not on type. This is different from the C++ `new` operator that returns a pointer whose type relies on the operand. One may "cast" this pointer to a specific type:

```
int *ptr;
ptr = malloc(10 * sizeof (*ptr));              /* without a cast */
ptr = (int *)malloc(10 * sizeof (*ptr));       /* with a cast */
ptr = reinterpret_cast<int *>(malloc(10 * sizeof (*ptr))); /* with a cast, for C++ */
```

src : https://stackoverflow.com/questions/20094394/why-do-we-cast-return-value-of-malloc

Question 3

Not yet answered

Marked out of 1.0

Flag question

Library dari fungsi malloc() tersimpan di dalam header file, yaitu .....

- mem.h
- stdlib.h
- math.h
- stdio.h
- memory.h

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL SQL CONSOLE GITLENS

```
Latihan1.c:3:1: note: include '<stdlib.h>' or provide a declaration of 'malloc'
    2 | #include <stdio.h>
  +++ |+#include <stdlib.h>
    3 |
Latihan1.c:8:16: warning: incompatible implicit declaration of built-in function 'malloc' [-Wbuiltin-declaration-mismatch]
    8 |     d1 = (int*)malloc(n*sizeof(int));
      |                ^~~~~~
Latihan1.c:8:16: note: include '<stdlib.h>' or provide a declaration of 'malloc'
```

Question 4

Not yet answered

Marked out of 1.0

Flag question

Instruksi yang salah untuk membuat array 1 dimensi berukuran n yang setiap elemennya diinisialisasi dengan nilai 0 adalah .....

```
int *dt;
dt=(int*)malloc(n*sizeof(int));
memset(dt, 0, (n*sizeof(int)));
```

```
int *dt={0};
```

```
int dt[n]={0};
```

```
int *dt;
dt=(int*)malloc(n*sizeof(int));
memset(dt, 0, (n*sizeof(dt[0])));
```

```
int *dt;
dt = (int*)calloc(n, sizeof(int));
```

Question 5

Not yet answered

Marked out of 1.0

Flag question

Perhatikan program berikut:

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

int main()
{
    int row = 3, col = 4, i, j, count;

    int (*arr)[col] = calloc(row, sizeof *arr);

    count = 0;
    for (i = 0; i < row; i++)
        for (j = 0; j < col; j++)
            arr[i][j] = ++count;

    printf("%d\n", arr[1][2]);

    free(arr);

    return 0;
}
```

Output dari program tersebut adalah ..... (Tuliskan jawaban Anda dengan tepat).

Answer:

Question 6

Not yet answered

Marked out of 1.0

Flag question

Time left 0:37:33

Berikut adalah kerangka program untuk membaca matrik 2 dimensi:

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

int* readMatrix(int *r, int *c);

int main()
{
    int row,col;
    int *dt;
    dt=readMatrix(&row, &col);
    .....
    .....
    return 0;
}
```

Implementasi fungsi readMatrix() yang tepat adalah .....

○

```
int* readMatrix(int *r, int *c)
{
    int nilai;
    int *mat;
    scanf("%d %d", r, c);
    mat=(int*)malloc(*r * *c * sizeof(int));
    for (int i=0; i<*r; i++)
    {
        for(int j=0; j<*c; j++)
        {
            scanf("%d", &nilai);
            mat[i][j]=nilai;
        }
    }
    return mat;
}
```

Time left 0:37:

○

```
int* readMatrix(int *r, int *c)
{
    int nilai;
    int *mat;
    scanf("%d %d", r, c);
    mat=(int*)malloc(r*c*sizeof(int));
    for (int i=0; i<*r; i++)
    {
        for(int j=0; j<*c; j++)
        {
            scanf("%d", &nilai);
            mat[i*(*c)+j]=nilai;
        }
    }
    return *mat;
}
```

○

```
int* readMatrix(int *r, int *c)
{
    int nilai;
    int *mat;
    scanf("%d %d", r, c);
    mat=(int*)malloc(r*c*sizeof(int));
    for (int i=0; i<*r; i++)
    {
        for(int j=0; j<*c; j++)
        {
            scanf("%d", &nilai);
            mat[i*(*c)+j]=nilai;
        }
    }
    return &mat;
}
```

Time left 0:37:26

○

```
int* readMatrix(int *r, int *c)
{
    int nilai;
    int *mat;
    scanf("%d %d", r, c);
    mat=(int*)malloc(r*c*sizeof(int));
    for (int i=0; i<r; i++)
    {
        for(int j=0; j<c; j++)
        {
            scanf("%d", &nilai);
            mat[i][j]=nilai;
        }
    }
    return mat;
}
```

○

```
int* readMatrix(int *r, int *c)
{
    int nilai;
    int *mat;
    scanf("%d %d", r, c);
    mat=(int*)malloc((*r)*(*c)*sizeof(int));
    for (int i=0; i<*r; i++)
    {
        for(int j=0; j<*c; j++)
        {
            scanf("%d", &nilai);
            mat[i*(*c)+j]=nilai;
        }
    }
    return mat;
}
```

Question 7

Not yet answered

Marked out of 1.0

Flag question

Instruksi:

```
ptr=calloc(m,n);
```

setara dengan .....

```
ptr=malloc(m*n);
memset(ptr, n, m);
```

```
ptr=malloc(m*n);
memset(ptr, m, n);
```

```
ptr=malloc(m,n);
memset(ptr, m, n);
```

```
ptr=malloc(m*n);
memset(ptr, 0, m*n);
```

```
ptr=malloc(n,m);
memset(ptr, n, m);
```

Time left 0:36:08

Question 8

Not yet answered

Marked out of 1.0

Flag question

Perhatikan program berikut:

```
#include <stdio.h>
void print(int *arr, int m, int n)
{
    int i, j;
    for (i = 0; i < m; i++)
    for (j = 0; j < n; j++)
        printf("%d ", *((arr+i*n) + j));
    printf("\n");
}

int main()
{
    int arr[3][4] = {{1, 2, 3, 4}, {5, 6, 7}, {8, 9}};
    print((int *)arr, 3, 4);
    return 0;
}
```

Output dari program tersebut adalah .....

Output dari program tersebut adalah .....

- `1 2 3 4 5 6 7 0 8 9 0 0`
- `1 2 3 4 5 6 7 8 9 0 0 0`
- `1 2 3 4 5 6 7 8 9 1 2 3`
- Tidak ada output, karena program mengalami error pada instruksi:
  `print((int *)arr, 3, 4);`
- `1 2 3 4 5 6 7 8 9 10 11 12`

Question 9

Not yet answered

Marked out of 1.0

Flag question

Mana saja fungsi-fungsi dalam pemrograman C yang berkaitan dengan alokasi memori secara dinamis?

- [ ] calloc()
- [ ] malloc()
- [ ] clear()
- [ ] realloc()
- [ ] stdlib()
- [ ] memory()
- [ ] free()
- [ ] memset()

Question 10

Not yet answered

Marked out of 1.0

Flag question

Diantara berikut, yang menimbulkan error untuk mendeklarasikan variabel array 2 dimensi adalah .....

```
#define N 100
int *arr[N];
```

```
int **arr;
```

```
int arr[][];
```

```
#define N 100
int arr[N][N];
```

```
#define N 100
int *arr[][N]={1,2,3,4,5,6,7,8,9};
```