Function Pointers Exmaple In C

Sayak Haldar
IIEST, Shibpur

Function pointer to a function which returns dynamically allocated 1-d integer array

```
#include <stdio.h>

#include<stdlib.h>

int * initialize();

int main()

{

        int *((*x)());//x is a pointer to a function which would return an integer array

        x=initialize;

        int *arr=x();

        printf("after x\n");

        int i;

        for(i=0;i<10;i++)

       {

            scanf("%d",&arr[i]);

        }

       printf("Printing the value stored in arr\n");

       for(i=0;i<10;i++)

      {

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

       }

     return 0;

}

int * initialize()

{
```

```
	int *arr;
	arr=(int *)malloc(10*sizeof(int));
	return arr;
}
```

Function pointer to a function which returns dynamically allocated 2-d array

```
#include<stdio.h>
#include<stdlib.h>
int ** initialize matrice(int row,int col)
{
	int **matrice;
	matrice=(int **)malloc(row*sizeof(int *));
	int i;
	for(i=0;i<row;i++)
	{
		matrice[i]=(int *)malloc(col*sizeof(int));
	}
      return matrice;
}
int main()
{
	int row,col;
	printf("Enter the row number:");
	scanf("%d",&row);
	printf("Enter the column number:");
```

```
	scanf("%d",&col);
	int **((*function_ptr)(int,int));
	function_ptr=initializematrice;
	int **matrice;
	matrice=function_ptr(row,col);
	int i,j;
	for(i=0;i<row;i++)
	{
		for(j=0;j<col;j++)
		{
			scanf("%d",&matrice[i][j]);
		}
	}
	printf("Printing the matrice\n");
	for(i=0;i<row;i++)
	{
		for(j=0;j<col;j++)
		{
			printf("%d ",matrice[i][j]);
		}
		printf("\n");
	}
	return 0;
}
```

A function pointer to a function which returns an array which has row as user input and has fixed column number as 5

```
#include<stdio.h>

#include<stdlib.h>

int (*initializematrice(int row))[5]

{

        int **matrice;

        matrice=(int **)malloc(row*sizeof(int *));

        int i;

        for(i=0;i<row;i++)

        {

                matrice[i]=(int *)malloc(5*sizeof(int));

        }

    return matrice;

}

int main()

{

        int row;

        printf("Enter the row number:");

        scanf("%d",&row);

        int (*(*function_ptr)(int))[5];

        function_ptr=initializematrice;

        int **matrice;

        matrice=function_ptr(row);

        int i,j;
```

```
	for(i=0;i<row;i++)
	{
		for(j=0;j<5;j++)
		{
			scanf("%d",&matrice[i][j]);
		}
	}
	printf("Printing the matrice\n");
	for(i=0;i<row;i++)
	{
		for(j=0;j<5;j++)
		{
			printf("%d ",matrice[i][j]);
		}
		printf("\n");
	}
	return 0;
}
```

This would eventually work properly though it gives two warnings.

Also:

```
int (*myfunc())[10]
{
		int (*ret)[10]=malloc(10*10*sizeof(int));
		return ret;
}
```

This function return pointer to 10 arrays of 10 ints.