

به نام خدا

دستور کار کارگاه کامپیوتر شماره ۷

توسعه برنامه‌های نسبتاً پیچیده کار با توابع و آرایه‌ها

در این دستور کار قرار است با استفاده از توابع و آرایه‌ها تعدادی برنامه نسبتاً پیچیده را پیاده‌سازی کنیم. در ادامه برنامه را توضیح خواهیم داد.

توجه: برای پیاده‌سازی هیچ کدام از توابع، اجازه ندارید از کتابخانه‌های آماده استفاده کنید!!!

۱. برنامه‌ای بنویسید که رشته‌ای را از ورودی خوانده، مشخص کند که آیا رشته از هر دو طرف که در نظر گرفته شود یکسان است یا خیر (مثلاً رشته‌ی "beeb" چنین خاصیتی دارد). در صورتی که رشته چنین خاصیتی نداشت، از کاربر دو کاراکتر ch1 و ch2 را دریافت کرده و تمامی کاراکترهای ch1 را در رشته به ch2 تبدیل کند.

```
#include <stdio.h>
#include <conio.h>
void replace (char[], char, char);
bool isMirrored (char[]);
int main() {
    char str[100], source_letter, target_letter;
    printf ("Enter the String: ");
    //Write Your Code Here
    _getch();
    return 0;
}
void replace(char string[], char source_letter, char target_letter){
    //Write Your Code Here
}

bool isMirrored(char string[]){
    //Write Your Code Here
    return mirrored ;
}
```

۲. برنامه ای بنویسید که در ابتدا از کاربر تعداد رشته‌های ورودی (n) را دریافت کند و سپس n رشته را از ورودی بگیرد، آنها را از نظر الفبا مقایسه و مرتب سازی شده (مانند Dictionary) در خروجی چاپ کند. در صورتی که کاربر پیش از وارد کردن تمامی n رشته، رشته "FiNiSh" را وارد کند برنامه باید دریافت ورودی را متوقف کند و پردازش را بر روی رشته‌های وارد شده انجام دهد. (پردازش بر روی رشته "FiNiSh" انجام نخواهد شد).

راهنمایی: برای پیاده‌سازی مرتب‌سازی، پیاده سازی را همانند مرتب سازی برای اعداد انجام دهید و در قسمت مقایسه دو رشته، نتیجه خروجی تابع compare را برای مقایسه در نظر بگیرید.

```
#include <stdio.h>

int compare(char first[], char second[]);
int checkFinish(char input[]);
void bubbleSort(char str[][]);

void main() {
    char str[100][100];
    int n;
    printf("Please enter number of strings(Max. 100): \n");
    scanf("%d", &n);
    //Write your code here!
}

int compare(char first[], char second[])
{
    //Write your code here!
    //Note: if first>second      => return 1
    //       if first==second    => return 0
    //       if first<second     => return -1
}
int checkFinish(char input[])
{
    //Write your code here!
}
void bubbleSort(char str[][]){
    //Write your code here!
}
```

۳. برنامه‌ای بنویسید که از کاربر سه مقدار: اسم، قد (واحد: متر) و وزن (واحد: کیلوگرم) را دریافت کند و برای تمامی اسامی معیار BMI^1 را محاسبه کند و به ترتیب صعودی معیار BMI، اسامی و BMI آن‌ها را در خروجی چاپ کند ($BMI = \frac{mass}{height^2}$). برنامه زمانی که کاربر مقدار “FiNiSh” را برای اسم وارد می‌کند، دریافت ورودی را متوقف کرده و شروع به انجام پردازش و نمایش خروجی می‌کند. راهنمایی: مقایسه را بر مبنای BMI انجام دهید. تفاوت در آنجاست که در هنگام جابجایی مقادیر در آرایه BMI برای مرتب سازی این آرایه، همزمان مقادیر مرتبط با آن در آرایه اسامی را نیز جابجا کنید.

```
#include <stdio.h>

float BMI(int mass, float height);
int checkFinish(char input[]);
void bubbleSort(char names[][100], float BMI[]);

void main() {
    char names[100][100];
    int mass[100];
    float height[100];
    float BMI[100];
    //Write your code here!
}

float BMI(int mass, float height)
{
    //Write your code here!
}
int checkFinish(char input[])
{
    //Write your code here!
}
void bubbleSort(char names[][100], float BMI[]){
    //Write your code here!
}
```

۴. برنامه‌ای بنویسید که در ابتدا یک عدد (n) را از کاربر دریافت کرده و سپس یک رشته name دریافت کند. حال تعداد n رشته را از کاربر دریافت کرده و مشخص کند آیا رشته name در هر یک از رشته‌های وارد شده ظاهر شده است یا خیر. در صورتی که ظاهر شده باشد محل ظهور اولین کاراکتر را ذخیره کند. در خروجی رشته‌هایی که name در آن‌ها باشد را به ترتیب صعودی محل ظهور name، نمایش دهد. به مثال زیر دقت کنید:

Input:

3

aut

ceitaut(tp)

polytechnic

author

Output:

author, 1

ceitaut(tp), 5

```
#include <stdio.h>
int CheckString(char input[], char name[]);
void bubbleSort(char names[][100], int location[]);

void main() {
    int n;
    char names[100][100];
    char strs[100][100];
    int location[100];
    //Write your code here!
}

int CheckString(char input[], char name[])
{
    //Write your code here!
    //Hint: you can return -1 if the input string does not contain the name
    string
}
void bubbleSort(char names[][100], int location[]){
    //Write your code here!
}
```

توجه: در تمامی برنامه‌ها، حداکثر تعداد رشته‌ها ۱۰۰ رشته و حداکثر طول هر رشته نیز ۱۰۰ کاراکتر خواهد بود.

توجه: برای نمایش خروجی در تمامی سوالات، باید یک تابع بنویسید که با توجه به عملیات انجام شده، خروجی مناسب را نمایش دهد. تابع خروجی به شکل زیر است:

```
void doPrint(List of Arguments){
    //Write your code here!
}
```