|  |
| --- |
| **KING SAUD UNIVERSITY**  **COLLEGE OF COMPUTER AND INFORMATION SCIENCES**  **COMPUTER SCIENCE DEPARTMENT** |

|  |  |  |
| --- | --- | --- |
| **KING SAUD UNIVERSITY**  **COLLEGE OF COMPUTER AND INFORMATION SCIENCES**  **COMPUTER SCIENCE DEPARTMENT** | | |
| **CSC215** | **Lab 6**  Functions & Memory Management | **2nd Semester 1436-1437** |

* Write a program that stores names of the best hospitals in Riyadh into an array of strings.
  + Define a constant variable MAX and make it equal to 5.
  + Use the main code provided in the end of this question. Which shows a menu where the user will choose one of the 4 options:
    - Add a new hospital name: Which calls AddName function.
    - Delete a hospital name: Which calls RemoveName function.
    - Print the hospitals names: Which calls PrintNames function.
    - Exit: Which terminates the program.
* Write the following functions:
  + Write the function ***AddName*** that takes an array of strings pointers called Names and a pointer of integer size.
    - Check if there is still enough space to store a new name.
    - Hint: you will need to use the value of MAX to check.
    - If there is a space, ask the user to input the hospital name and store it in a huge array of char (70 char).
    - Calculate the length of the hospital name.
    - Allocate a dynamic memory to store the entered hospital name and store its location in one of Names indexes.
    - Increment the size by one.
      * void AddName(char \*Names[],int \*size)
  + Write the function ***RemoveName*** that takes an array of strings pointers called Names and a pointer of integer size.
    - Check if the array is not empty.
    - If it’s not, asks the user to input the index of the hospital name that he wants to delete. Assume that the user will enter indices starting from 0.
    - If the entered index is within a correct range of indices, Free the dynamically allocated memory.
    - Shift left all the hospitals names that comes after it.
    - Decrement the size by one.
      * void RemoveName(char \*Names[],int \*size)
  + Write the function ***PrintNames*** that takes an array of strings pointers called Names and an integer size. Then prints all of the names separated by commas (,).
    - Hint: Make sure that the array is not empty before printing.
    - void PrintNames(char \*Names[],int size)

main()

{

**char** \*Names[MAX];

**int** size = 0;

**int** c;

**do**{

printf(**"=========================\n"**);

printf(**"1- Add a new name.\n"**);

printf(**"2- Delete an old name.\n"**);

printf(**"3- Print names.\n"**);

printf(**"4- Exit.\n"**);

printf(**"=========================\n"**);

printf(**"Enter your choice: "**);

scanf(**"%i"**, &c);

printf(**"=========================\n"**);

**switch**(c){

**case** 1: AddName(Names,&size);

**break**;

**case** 2: RemoveName(Names,&size);

**break**;

**case** 3: PrintNames(Names,size);

**break**;

**case** 4: printf(**"Good bye.\n"**);

**break**;

default: printf(**"ERROR: Bad input.\n"**);

}

}**while**(c != 4);

}

Name your file using the following naming convention:

* “Lab6\_YourFirstName\_YourLastName.c"
* Don’t forget to move to your own directory
* Compile your code and execute it.

**Model Answer:**

#include <stdio.h>  
#include <string.h>   
#include <stdlib.h>  
#define MAX 5  
void AddName(char \*[],int \*);  
void RemoveName(char \*[],int \*);  
void PrintNames(char \*[],int);  
  
int main()

{

char \*Names[MAX];

int size = 0;

int c;

do{

printf("=========================\n");

printf("1- Add a new name.\n");

printf("2- Delete an old name.\n");

printf("3- Print names.\n");

printf("4- Exit.\n");

printf("=========================\n");

printf("Enter your choice: ");

scanf("%i", &c);

printf("=========================\n");

switch(c){

case 1:   
 AddName(Names,&size);

break;

case 2:   
 RemoveName(Names,&size);

break;

case 3:   
 PrintNames(Names,size);

break;

case 4:   
 printf("Good bye.\n");

break;

default:   
 printf("ERROR: Bad input.\n");

}

}while(c != 4);

}

void AddName(char \*Names[],int \*size)  
{  
 int Copysize = \*size;  
 char \*s;  
 if (Copysize >= MAX)  
 printf("\n ERROR: Array is full. Cannot add.");  
 else  
 {  
 int i,length=0;  
 char name[100];  
 printf("Enter the name: ");  
 scanf("%s",name);  
 for(i=0; name[i]!='\0'; i++)  
 length++;  
 s = (char \*)malloc((length+1)\*sizeof(char));  
 strcpy(s, name);  
 Names[\*size]=s;  
 \*size=\*size+1;  
 printf("\n The entered data has been added successfully.\n");  
 } }  
 void RemoveName(char \*Names[],int \*size)  
{  
if (\*size == 0)  
printf("There are no data to delete");  
else  
{  
int index, i;  
printf("Please Enter the index of the element you want to delete starting from 0 ");  
scanf("%d", &index);  
if(index<0 || index >= MAX){  
printf("The entered index is incorrect");  
return;  
}  
if(index >= \*size){  
printf("The entered index is already free, There is nothing to be deleted");   
return;  
}  
free(Names[index]);  
for (i=index; i<MAX-1; i++)  
{  
Names[i] = Names[i+1];  
}[MAX-1] = NULL;  
\*size= \*size-1;  
printf("Deletion is done successfully");  
}  
}  
  
void PrintNames(char \*Names[],int size)  
{  
int i;  
if (size>0){  
for(i=0; i<size; i++)  
printf("%s , ",\*(Names+i));("\n");  
}  
else  
{  
printf("There are No data to print");  
printf("\n");  
}  
}