In The Name Of God

The Most Compassionate And Merciful

Problem Set - 7

Chapter 5 - Pointers

------------------------------------------------

|  |
| --- |
| 1. */\* strcpy: copy t to s\*/* void strcpy(char \*s, char \*t); |

2.

|  |
| --- |
| */\* strcmp: return <0 if s<t, 0 if s==t, >0 if s>t \*/* int strcmp(char \*s, char \*t); |

3.

|  |
| --- |
| write a pointer version of the function strcat that we showed in Chapter 2: strcat(s,t) copies the string t to the end of s. |

4.

|  |
| --- |
| write the function strend(s,t), which returns 1 if the string t occurs at the end of the string s, and zero otherwise. |

5.Run this script and try to understand it! It helps you to implement 2d array using pointer to pointer.

It may help you in midterm and final exam!

|  |
| --- |
| int array[5][5];  int \*\* parray;  parray = (int \*\*)malloc(sizeof(int \*) \* 5);   int i,j;   for(i = 0 ; i<5 ; i++)  for(j=0 ; j<5 ; j++)  array[i][j] = i\*j;    printf("\n\n\n ARRAY : \n\n\n");  for(i = 0 ; i<5 ; i++){  printf("\n");  for(j=0 ; j<5 ; j++)  printf("%d\t" , array[i][j]);  }     for(i = 0 ; i<5 ; i++){  \*(parray + i) = (int \*)malloc(sizeof(int) \* 5);  for(j=0 ; j<5 ; j++){  \*(\*(parray + i) + j) = j+i\*2;  }  }    printf("\n\n\n PARRAY : \n\n\n");  for(i = 0 ; i<5 ; i++){  printf("\n");  for(j=0 ; j<5 ; j++)  printf("%d\t" , \*(\*(parray + i) + j));  } |

6.Try to implement 3D array using pointer to pointer to pointer!

(check previous example it may helps you)

|  |
| --- |
| \*\*\*array; |

7.Write a function with below prototype which can return the biggest element of an array.

|  |
| --- |
| int find\_the\_biggest(int \* array , int array\_size); |

8.Try to write a function called realloc which can allocate more space and also save previous data.

|  |
| --- |
| int \* realloc( int \*previous\_array , int size\_of\_previous\_array , int new\_size );  int \* array; array = (int \*) malloc(sizeof(int) \* 5); array[0] = 1; array[1] = 1; array[2] = 1; array[3] = 1; array[4] = 1;  int \* newP = realloc( array , 5 , 10 ) ;  *// now array can store 5 more data!* |

9.Write function called print\_string() . It can print a string which get by pointer.

|  |
| --- |
| char \* str = "Dr. Ali Hamzeh"; void print\_string(char \* str); *// output : Dr. Ali Hamzeh* |

10.write char \* to\_lower(char \* str); function . It will change uppercase letters to lowercase.

|  |
| --- |
| char \*str = "We Have a Quiz NeXT WEEK :(". str = to\_lower(str); *// be careful to\_lower(&str) is wrong!*  *// str = we have a quiz next week :(* |

**Solution Q10 :**

|  |
| --- |
| #include <stdio.h>  char \* to\_lower(char \* text){  int i = 0;  char \* new\_text;  while(\*(text + i) != '\0'){  if(\*(text + i) >= 'A' && \*(text + i) <= 'Z'){  \*(new\_text + i) = \*(text + i) + 32;  }  else{  \*(new\_text + i) = \*(text + i);  }  i++;  }  return new\_text; }  int main(){  char \* text = "Hello World!";  text = to\_lower(text);  printf("%s" , text); } |

11. write a program which can get unlimited numbers from user and when user press EOF , it will calculate the average of numbers. ( malloc is your friend! )

Hint : first consider we have 10 space , if user enter more than 10 numbers you should use reallocation.