In The Name Of God

The Most Compassionate And Merciful

Problem Set - 8

Linked List

1. Develop a linked list then try to implement these functions for your linked list.(Simple)

|  |
| --- |
| Struct Node{  int data;  Struct Node \* next; }; |

|  |
| --- |
| 1.struct Node \* insert\_at(int value , struct Node \*head , int house); 2.struct Node \* edit(int house, int newValue,struct Node \*head); 3.struct Node \* delete\_by\_index(int index , struct Node \*head); 4.struct Node \* delete\_by\_value(int number , struct Node \*head); 5.int search(int number , struct Node \*head); 6.struct Node \*add\_end(int data , struct Node \*head); 7.struct Node \*add\_first(int newValue , struct Node \*head); 8.struct Node \*delete\_first(struct Node \*head); 9.struct Node \*delete\_last(struct Node \*head); 10.void delete\_list(struct Node \*head); 11.int get(int house , struct Node \*head); 12.int listlen(struct Node \*head); 13.void bubbleSort(struct Node \*head); 14.void printList(struct Node \*head); |

// Given To you in the bonus part .

3. Develop a doubly linked list and implement previous functions for the linked list.(Double)

|  |
| --- |
| 1.void printList(struct Node \*head);  2.int listlen(struct Node \*head);  3.struct Node \* insert\_at(struct Node \*head ,int value , int index);  4.int search( struct Node \*head, int number ); 5.struct Node \*add\_end(struct Node \*head, int data ); 6.struct Node \*add\_first(struct Node \*head, int newValue); 7.struct Node \*delete\_first(struct Node \*head); 8.struct Node \*delete\_last(struct Node \*head); 9.void delete\_list(struct Node \*head); 10.int get( struct Node \*head, int index ); 11.struct Node \* edit(struct Node \*head, int index, int newValue); 12.struct Node \* delete\_by\_index(int index , struct Node \*head); 13.struct Node \* delete\_by\_value(int number , struct Node \*head); |

4. Develop a **circular** linked list and implement previous functions for it.

|  |
| --- |
| Struct Node{  int data;  Struct Node \* next; }; |

1. Write a program that gets unlimited integer inputs from the user and calculates the average of them .
2. rewrite the following functions with the given prototype:
   1. void \* realloc ( void \* buffer , int size )
   2. int strCmp ( char \* sentence1 , char \* sentence2 )
   3. void Print ( int \* numbers , int size)
   4. int \* Pallindrom\_finder ( int \* numbers , int size)
   5. void swap( int \* a , int \* b )
   6. char \* strCat(char \* sen1 , char \* sen2 )
   7. void strCpy ( char \* to , char \* from )
3. Write a program that gets unlimited names and saves them into one array like below :(array of pointers )

Char \* name

Char \*\* arrayList\_Names

1. Write a program that gets 3 coordinates from the user and determines if we can create a triangle with those coordinates or not .( cord : x & y )

Typedef Struct cord {

Int x , y;

}Cordination ;

1. Write a program that the user writes a word in english and the program shows its synonym in persian ( a dictionary ) { you have to define an array of pointers to structures }
2. Write a program that gets 4 coordinates as 4 points that make a rectangle and determines the surface of the rectangle.
3. Write a program that gets ” n “ as input (number of students ) and gathers the information of each student as below :

struct student{

Char name[20] ;

Char last\_name[20];

Int age ;

Int grades[20];

};

Void main(void){

….

Struct student all\_students[n];

….

}

The program should have the following abilities :

1 - calculate the average of the scores.

2-sort the students by their average.