In The Name Of God

The Most Compassionate And Merciful

Problem Set 9

BTree & Hash Table

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

Sheet info :

* Problem Set 9
* Due Date : Not Set
* Just upload scorable questions on HWS.
* File name format : “**StudentNumber\_PS9.zip**”
* Do not hesitate to ask any question from your graders!

1. Develop a BTree with two data and three children & implements below functions for it.

|  |
| --- |
| struct Node{  int lower\_data;  int bigger\_data;  struct Node \*left;  struct Node \*right;  struct Node \*middle;  int data\_counter; }; |

|  |
| --- |
| struct Node \*insert(struct Node \*root , int data); void free\_btree(struct Node \*root); void print\_btree(struct Node \*root); struct Node \* search(struct Node \*root , int data); int count(struct Node \*root , int counter); struct Node \* delete(struct Node \*root , int data);  Struct Node \*balance ( struct Node \* root ) ; |

2. Develop a BTree with three data and four children & implements same functions as above for it.

|  |
| --- |
| struct Node{  int lower\_data;  int middle\_data;  int bigger\_data;  struct Node \*left\_left;  struct Node \*left\_middle;  struct Node \*right\_middle;  struct Node \*right\_right;  int data\_counter; }; |
|  |

|  |
| --- |
| struct Node \*insert(struct Node \*root , int data); void free\_btree(struct Node \*root); void print\_btree(struct Node \*root); struct Node \* search(struct Node \*root , int data); int count(struct Node \*root); struct Node \* delete(struct Node \*root , int data); |

3.Develop a Hash-Table and implement below functions for it.

|  |
| --- |
| struct Node{  int data;  struct Node \*next; };   int main(int argc, char const \*argv[]){  struct Node \* array[10] = {NULL};  return 0; } |

|  |
| --- |
| void hash\_add(int number , struct Node \* array[]); struct Node \* hash\_search(int number , struct Node \* array[]);  Void hash\_delete(int number , struct Node \* array[]);  void hash\_print(struct Node \* array[]); |