DPPL LAB ASSIGNMENT – 2

HEAP MANAGEMENT ALGORITHMS

TEAM MEMBERS:-

1. Anukalp Pandey (BT20CSE144)
2. Anubhav Govind (BT20CSE113)
3. Utkarsh Srivastava (BT20CSE139)

STRUCTURE OF NODE:-

```
struct node
{
    int status;     // Meta
    int size;       // Meta
    node *next;     // Pointer
    node *prev;     // Pointer
    long long *num; // Payload
};
```

The structure of node is shown above:-

- Status variable stores the allocation status of the block.
  Status = =0 → block is not allocated
  Status == 1 → block is allocated.

- Size variable stores the size of the block(META_DATA + Payload).
- *next, *prev are pointers to next and previous blocks in the Heap list.
- long long *num is the payload corresponding to the heap block.

GLOBAL VARIABLES:-

```
node *heap = NULL;

node *forNextFit = heap;
node *bestFit = NULL;
int diff = INT_MAX;
```

- *heap is the pointer to the start of the Heap List.

- *forNextFit is the pointer to the block from which the implementation of next fit Algorithm will start.
- *bestFit is the pointer which stores the address of best block to be allocated at that instant.
- diff variable stores the difference between payload of best Size block and Size required at that instant.

MAIN MODULES:-

- Coalesce() -> This function merges two or more consecutive empty blocks in the heap list.
- Splitting()-> This function splits the block into two blocks if the difference between size of the initial block and the required block is more than the META_SIZE,
- freeMem() -> It changes the allocation status of the block and further calls the coalesce function to merge the adjacent blocks(if empty).
- allocateFirstFit() -> It allocates the block in the heap list in accordance with the First-Fit Algorithm.
- allocateNextFit() -> It allocates the block in the heap list in accordance with the Next-Fit Algorithm.
- allocateBestFit() -> It allocates the block in the heap list in accordance with the Best-Fit Algorithm.

UTILITY MODULES:-

- createNode() -> It creates the node for the heap block.
- createHeap() -> An array is passed to it with payload values and it creates the heap list in accordance with the values given.

- displayHeap() -> It displays the heap.