```
#include <stdio.h>


int g = 0, k = 0;

{
struct free {
    int tag;
    int size;
    struct free* next;
}* free_head = NULL, *prev_free = NULL;


struct alloc {
    int block_id;
    int tag;
    int size;
    struct alloc* next;
}* alloc_head = NULL, *prev_alloc = NULL;


void create_free(int c)
{
    struct free* p = (struct free*)
        malloc(sizeof(struct free));
    p->size = c;
    p->tag = g;
    p->next = NULL;
    if (free_head == NULL)
        free_head = p;
    else
        prev_free->next = p;
    prev_free = p;
    g++;
}


void print_free()
{
    struct free* p = free_head;
    printf("Tag\tSize\n");
    while (p != NULL) {
        printf(" p->tag\t");
         printf(" p->size \n");
        p = p->next;
    }
```

```
}


void print_alloc()
{
    struct alloc* p = alloc_head;
    printf("Tag\tBlock ID\tSize\n");
    while (p != NULL) {
        printf(" p->tag \t ");
           printf(" p->block_id \t\t");
            printf(" p->size \n");
        p = p->next;
    }
}


void create_alloc(int c)
{

    struct alloc* q = (struct alloc*)
        malloc(sizeof(struct alloc));
    q->size = c;
    q->tag = k;
    q->next = NULL;
    struct free* p = free_head;


    while (p != NULL) {
        if (q->size <= p->size)
            break;
        p = p->next;
    }


    if (p != NULL) {

        q->block_id = p->tag;
        p->size -= q->size;
        if (alloc_head == NULL)
            alloc_head = q;
        else {
            prev_alloc = alloc_head;
            while (prev_alloc->next != NULL)
                prev_alloc = prev_alloc->next;
            prev_alloc->next = q;
        }
        k++;
    }
```

```
    else
       printf("Block of size ",c);
           printf(" can't be allocated\n");
}


void delete_alloc(int t)
{

    struct alloc *p = alloc_head, *q = NULL;


    while (p != NULL) {
        if (p->tag == t)
            break;
        q = p;
        p = p->next;
    }
    if (p == NULL)
        printf( "Tag ID doesn't exist\n");
    else if (p == alloc_head)
        alloc_head = alloc_head->next;
    else
        q->next = p->next;
    struct free* temp = free_head;
    while (temp != NULL) {
        if (temp->tag == p->block_id) {
            temp->size += p->size;
            break;
        }
        temp = temp->next;
    }
}

}
int main()
{
    int blockSize[] = { 100, 500, 200 };
    int processSize[] = { 417, 112, 426, 95 };
    int m = sizeof(blockSize)
            / sizeof(blockSize[0]);
    int n = sizeof(processSize)
            / sizeof(processSize[0]);

    for (int i = 0; i < m; i++)
        create_free(blockSize[i]);

    for (int i = 0; i < n; i++)
```

```
        create_alloc(processSize[i]);

    print_alloc();


    delete_alloc(0);

    create_alloc(426);
    printf("After deleting block with tag id 0.\n");

    print_alloc();
}
```