

MIS: 612303017

NAME: Aman Morghade

TYPE: Assignment 2 – ASCII ADT

ascii.h

```
typedef struct node {  
    int digit;  
    struct node* next;  
    struct node* prev;  
} node;
```

```
typedef struct ascii {  
    node* front;  
    node* rear;  
} ascii;
```

*// Helper Functions*

```
int isEmpty(ascii* l);  
void insertAtBeginning(ascii *l, int digit);  
void removeAtBeginning(ascii* l);
```

*// ASCII Prototypes*

```
void init_ASCII(ascii** l);  
void ASCII_of(ascii* l, char c);  
void traverse(ascii* l);  
void destroy(ascii* l);
```

ascii.c

```
#include <stdio.h>  
#include <stdlib.h>  
#include "ascii.h"
```

*// Function to check if the list is empty*

```
int isEmpty(ascii *l) {  
    return l->front == NULL;  
}
```

*// This functions initializes the ASCII ADT by taking the address of pointer to the ADT and setting it to NULL*

```
void init_ASCII(ascii** l) {  
    *l = (ascii*) malloc(sizeof(ascii));  
    (*l)->front = NULL;  
    (*l)->rear = NULL;  
}
```

*// Inserts a node(Digit) at the beginning of the list and handles the edge cases when list is empty and shifts front/rear accordingly*

```
void insertAtBeginning(ascii *l, int digit) {  
    node* p = (node*) malloc(sizeof(node));  
    p->digit = digit;  
  
    if(isEmpty(l)) {  
        l->front = p;  
        l->rear = p;  
        p->prev = NULL;  
        p->next = NULL;  
    } else {  
        p->prev = NULL;  
        p->next = l->front;  
        l->front->prev = p;  
        l->front = p;  
    }  
}
```

*// Returns void if the list is empty, else removes a node from the beginning of the list and shifts front accordingly*

```
void removeAtBeginning(ascii* l) {
```

```
    if(isEmpty(l)) return;
```

```
    node* p = l->front;
```

```
    l->front = l->front->next;
```

```
    if (l->front != NULL) {
```

```
        l->front->prev = NULL;
```

```
    } else {
```

```
        l->rear = NULL;
```

```
    }
```

```
    free(p);
```

```
}
```

*/\*\* This functions typecasts the character into it's ASCII Value, then iteratively the value is taken as it's modulo by 10*

*to retrieve the last digit of the value and inserts the value at the beginning of the list \*\*/*

```
void ASCII_of(ascii *l, char c) {
```

```
    int ascii_value = (int) c;
```

```
    if (ascii_value == 0) {
```

```
        insertAtBeginning(l, 0);
```

```
        return;
```

```
    }
```

```
    while(ascii_value) {
```

```
        int rem = ascii_value % 10;
```

```
        insertAtBeginning(l, rem);
```

```
        ascii_value /= 10;
```

```
    }  
}
```

*// Simply traverses the list iteratively, printing the digit of each node*

```
void traverse(ascii *l) {  
    for(node* p = l->front; p != NULL; p = p->next) {  
        printf("%d", p->digit);  
    }  
}
```

*// Iteratively calls the removeAtBeginning function until the list is Empty*

```
void destroy(ascii *l) {  
    while(!isEmpty(l)) {  
        removeAtBeginning(l);  
    }  
}
```

main.c

```
#include <stdio.h>
```

```
#include "ascii.c"
```

```
int main() {  
    ascii *l;  
    init_ASCII(&l);  
    ASCII_of(l, 'a');  
    traverse(l);  
    destroy(l);  
    traverse(l);  
    return 0;  
}
```

OUTPUT:

Tested for the input: char 'a'

```
PS C:\Users\Aman Morghade\OneDrive\Documents\DSA_CoEP\Assignment_2> gcc .\main.c
PS C:\Users\Aman Morghade\OneDrive\Documents\DSA_CoEP\Assignment_2> .\a.exe
97
PS C:\Users\Aman Morghade\OneDrive\Documents\DSA_CoEP\Assignment_2> |
```