Challenge H1: Sort an array of 0's, 1's and 2's in linear time complexity
Solution:

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

//function to swap by reference
void swap(int*a,int*b){
	int temp;
	temp=*b;
	*b=*a;
	*a=temp;
	return;
}
int* asort(int *a,int n){
	int low=0,mid=0,high=n-1;   //variables are set

	while(mid<=high){
		switch(a[mid]){
			case 0:    //if a[mid]==0
				//swap a[low] & a[mid], swapping by reference
				swap(&a[low],&a[mid]);
				low++;      //increment low
				mid++;      //increment mid
				break;
			case 1:    //if a[mid]==1
				mid++;      //increment mid
				break;
			case 2:    //if a[mid]==2
				//swap a[mid] & a[high], swapping by reference
				swap(&a[mid],&a[high]);
				high--;     //decrement high
				break;
		}
	}
	//returning adress of array(sorted)
	return a;
}

int main() {
	int n;

	printf("enter no of array elements\n");
	//input array length
	scanf("%d",&n);

	int* a=(int*)malloc(sizeof(int)*n);

	printf("enter array elements\n");
	//input array elements
	for(int j=0;j<n;j++)
		scanf("%d",&a[j]);
```

```
        //array is modified
        a=asort(a,n);
        printf("after being sorted.............\n");
        //printing the sorted array
        for(int j=0;j<n-1;j++)
                printf("%d ",a[j]);

        printf("%d\n",a[n-1]);

        return 0;
}
```
Output

```
First run:
enter no of array elements
10
enter array elements
0
2
2
1
1
1
2
1
2
1
after being sorted.............
0 1 1 1 1 1 2 2 2 2


Second run:
enter no of array elements
10
enter array elements
2
2
2
2
0
0
0
0
1
1
after being sorted.............
0 0 0 0 1 1 2 2 2 2
```

Challenge H2: Reverse a single linked list
Solution:

```
#include<bits/stdc++.h>

using namespace std;

class node{
	public:
		int data; // data field
		node *next;
};

node* reverse(node* head){
	node *next=NULL,*cur=head,*prev=NULL; //initialize the pointers
	while(cur!=NULL){//loop till the end of linked list
		next=cur->next;//next = cur->next to store the rest of the list;
		cur->next=prev;//change the direction of linked list
		prev=cur; //update prev
		cur=next; //update cur
	}

	head=prev; //update head
	return head; //return head
}

void traverse(node* head){
	node* current=head; // current node set to head
	int count=0; // to count total no of nodes
	printf("\ntraversing the list\n");
	while(current!=NULL){ //traverse until current node isn't NULL
		count++; //increase node count
		printf("%d ",current->data);
		current=current->next; // go to next node
	}
	printf("\ntotal no of nodes : %d\n",count);
}

node* creatnode(int d){
	node* temp=(node*)malloc(sizeof(node));
	temp->data=d;
	temp->next=NULL;
	return temp;
}

int main(){
	printf("creating the linked list by inserting new nodes at the begining\n");
	printf("enter 0 to stop building the list, else enter any integer\n");
	int k,count=1,x;

	node* curr;
	scanf("%d",&k);
	node* head=creatnode(k); //buliding list, first node
	scanf("%d",&k);

	//inserting at begining////
	while(k){
		curr=creatnode(k);
		curr->next=head;   //inserting each new node at the begining
		head=curr;
		scanf("%d",&k);
	}
	traverse(head); // displaying the list

	cout<<"reversing the list..........."<<endl;
	head=reverse(head);// reverse the linked list
	traverse(head);//display reversed linked list

	return 0;
}
```

Output

```
creating the linked list by inserting new nodes at the begining
enter 0 to stop building the list, else enter any integer
6
7
8
9
4
3
3
1
0

traversing the list
1 3 3 4 9 8 7 6
total no of nodes : 8
reversing the list...........

traversing the list
6 7 8 9 4 3 3 1
total no of nodes : 8
```