SOLUTIONS

Round 1:

Prob-1:

```
#include <iostream>

int n = printf("Hello World");

int main() {}
```

Prob-2:

```
#include <iostream>
using namespace std;

int main() {
        int n=4; // no. of soldiers
        int a[] = { 11, 12, 13, 14 };
        int lucky=0,unlucky=0;
        //for(int i=0;i<n;i++){
          //store no. of weapons by particular soldier in an array
          //cin>>a[n];
        //}

        for(int i=0;i<n;i++)
        {
          if(a[n]%2==0)  //condition for checking whether no. of weapons even or
not
                lucky++; // if satisfies increase lucky count by one
         else
                unlucky++; //else increase unlucky
        }

        if(lucky>unlucky)
          cout<<"READY FOR BATTLE"<<endl;  //display as per the required o/p
        else
          cout<<"NOT READY"<<endl;
        return 0;
 }
```

Prob-3:

```
#include <stdio.h>

// 59 is an ASCII value of the semicolon
#define SEMICOLON 59

int main()
{
   if (printf("%c", SEMICOLON)) {}
   // if (putchar(SEMICOLON)) {}
}
```

Prob-4:

```
int add(int a=-2, int b=-3) {
   return a-(-b);
}
```

Round 2:

Prob-1:

```
#include <bits/stdc++.h>
using namespace std;


int main() {
        // your code goes here
        int t;cin>>t;
        while(t--)
        {
           int n;cin>>n; int a[n]; int sum=0;
             for(int i=0;i<n;i++)
             {
               cin>>a[i];
                sum=sum+a[i];
             }
             if (n%2==0)
             {cout<<abs(sum)/2<<endl;}
             else
             cout<<"-1"<<endl;
        }
        return 0;
}
```

Prob-2:

```
#include <bits/stdc++.h>
using namespace std;

// Function to return the maximum water that can be stored
int maxWater(int arr[], int n)
{
	// To store the maximum water that can be stored
	int res = 0;
	// For every element of the array
	for (int i = 1; i < n - 1; i++) {
		// Find the maximum element on its left
		int left = arr[i];
		for (int j = 0; j < i; j++)
			left = max(left, arr[j]);
		// Find the maximum element on its right
		int right = arr[i];
		for (int j = i + 1; j < n; j++)
			right = max(right, arr[j]);
		// Update the maximum water
		res = res + (min(left, right) - arr[i]);
	}
	return res;
}

// Driver code
```

```
int main()
{
	int arr[] = { 0, 1, 0, 2, 1, 0, 1, 3, 2, 1, 2, 1 };
	int n = sizeof(arr) / sizeof(arr[0]);

	cout << maxWater(arr, n);

	return 0;
}
```

Round 3:

Prob: