

Esc101 : Practice Sheet 3

Question 1 - Minimum

An array of 10 integers is given. Complete the given program to find the minimum of the values of the numbers in the array.

```
#include <stdio.h>
int GetMin( _____ )
{
    int minval=_____;    //Stores the minimum value
    int i;
    for(i=0;i<10;i=i+2)
    {
        if(minval>_____)
            _____=_____;
    }
    return minval;
}
int main()
{
    int arr[10],temp,i;
    for(i=0;i<10;i++)
        scanf("%d",&arr[i]);    //Reads input
    for(i=0;i<10;_____)    //Preprocessing
    {
        if(arr[i]>arr[i+1])
        {
            temp=arr[i];
            arr[i]=arr[i+1];
            arr[i+1]=temp;
        }
    }
    printf("%d\n",_____);    //Prints the output
    return 0;
}
```

Question 2 – Missing Element

An array consists of numbers from 1 to n. Suppose one number goes missing. Find that number. The new array has only n-1 elements in any random order.

```

#include <stdio.h>
int Sum( _____ ) // Calculates sum of array elements
{
    int i,sum=_____;
    for(i=0;i< n-1;i++)
        sum+=_____;
    return sum;
}
int main()
{
    int arr[1000],i;
    scanf("%d",&n);
    for(i=0;i< n-1;i++) // Reads Input
        scanf("%d",&arr[i]);
    printf("%d\n", _____ ); // Prints the missing number
    return 0;
}

```

Question 3 – Sieve of Eratosthenes

Given a number n, print all primes smaller than or equal to n. It is also given that n is a small number.

For example, if n is 20, the output should be 2, 3, 5, 7, 11, 13, 17, 19

The sieve of Eratosthenes is one of the most efficient ways to find all primes smaller than n.

Eratosthenes' method:

1. Create a list of consecutive integers from 2 to n: (2, 3, 4, ... n).
2. Initially, let p equal 2, the first prime number.
3. Starting from p, count up in increments of p and mark each of these numbers greater than p itself in the list. These numbers will be 2p, 3p, 4p, etc.; note that some of them may have already been marked.
4. Find the first number greater than p in the list that is not marked. If there was no such number, stop. Otherwise, let p now equal this number (which is the next prime), and repeat from step 3.
5. When the algorithm terminates, all the numbers in the list that are not marked are prime.

```

void markMultiples(bool arr[], int a, int n)
{
    int i = 2, num;
    while ( (num = _____ ) <= n )
    {
        arr[ _____ ] = 1;
        ++i;
    }
}

```

```

void SieveOfEratosthenes( _____ )
{
    if (n >= 2)
    {
        bool arr[n];
        int i ;
        for (i=0; i<n; ++i)
        {
            arr[i] = _____ ;
        }

        for ( _____ ; i<n; ++i)
        {
            if ( .... )
            {
                printf("%d ", _____ );
                markMultiples( _____ , _____ , _____ );
            }
        }
    }
}

```

```

int main()
{
    int n = 100;
    SieveOfEratosthenes(n);
    return 0;
}

```