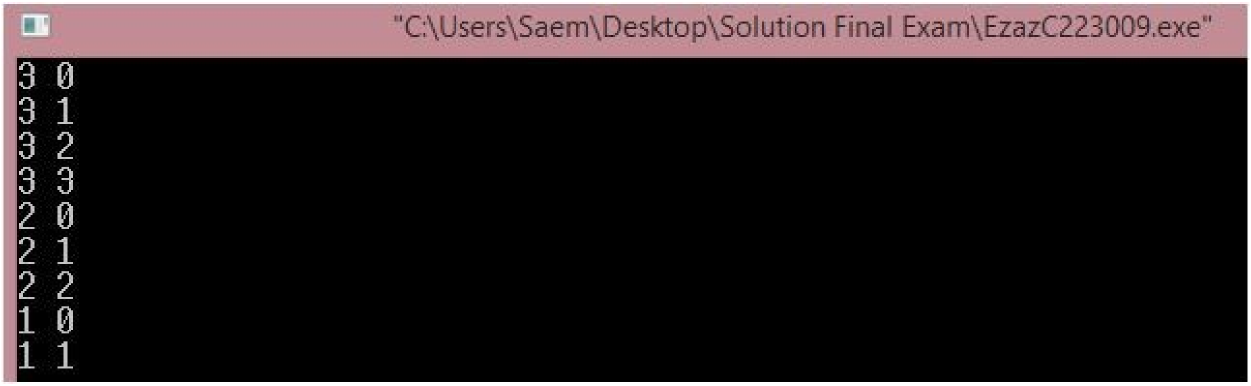
CSE Final Autumn 22 Solution

Name:Ezaz Ahmed ID:C223009 Section:AM

Hello Juniors

1(a):

```
#include<stdio.h>
int main()
{
    int i,j;
    for(i=3;i>=1;i--)
        for(j=0;j<=i;j++)
        printf("%d %d\n",i,j);
    return 0;
}
```



"C:\Users\Saem\Desktop\Solution Final Exam\EzazC223009.exe"

```
3 0
3 1
3 2
3 3
2 0
2 1
2 2
1 0
1 1
```

1(b):

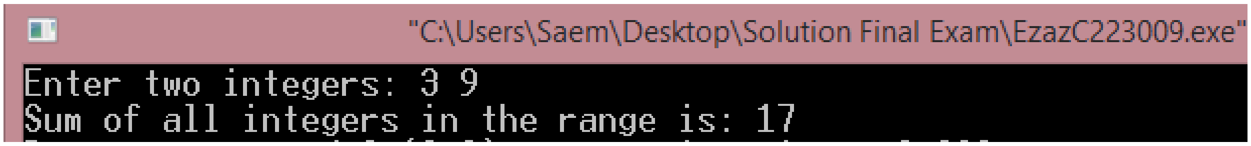
```
#include <stdio.h>
int main()
{
    int p, q;
    scanf("%d %d", &p, &q);
    for (int i = p; i <= q; i++)
    {
        if (i % 3 == 0)
        {
            continue;
        }
        printf("%d\n", i);
    }
return 0;
}
```

If we use break here:

If the **break** statement is encountered inside the loop when **i** is divisible by 3, it immediately terminates the loop, skipping any remaining iterations and the subsequent numbers will not be printed.

1(c):

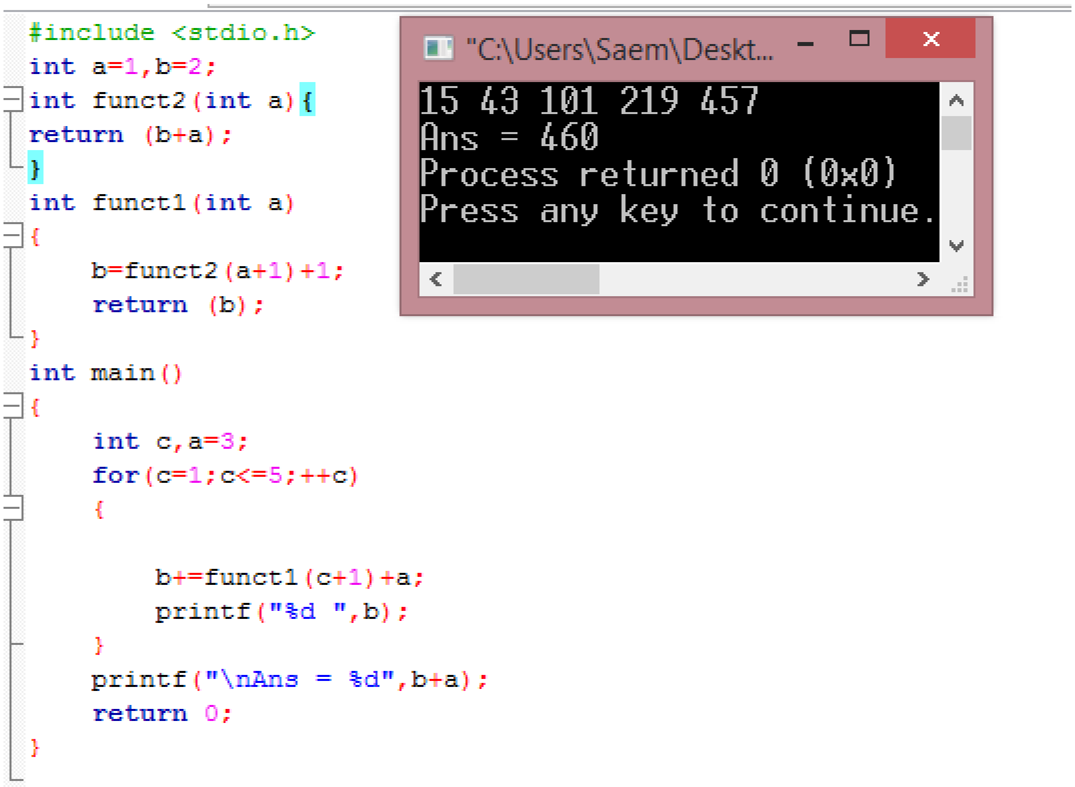
```
#include <stdio.h>
int main()
{
    int a, b;
    printf("Enter two integers: ");
    scanf("%d %d", &a, &b);
    int sum = 0;
    for (int i = a; i <= b; i++)
    {
        if(i%2!=0)
        {
            if (i % 3 == 0 || i % 5 == 0)
            {
                sum += i;
            }
        }
    }
    printf("Sum of all integers in the range is: %d",sum);
    return 0;
}
```



"C:\Users\Saem\Desktop\Solution Final Exam\EzazC223009.exe"

```
Enter two integers: 3 9
Sum of all integers in the range is: 17
```

2(a):



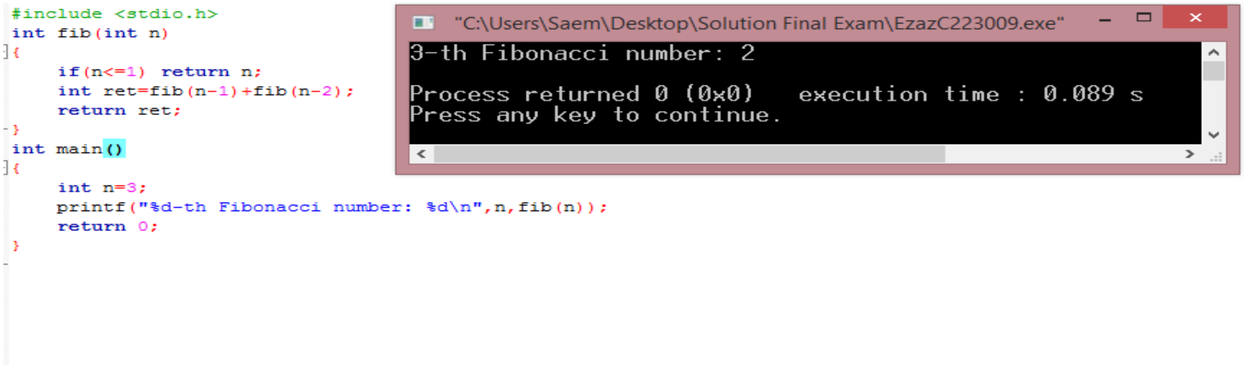
```c
#include <stdio.h>
int a=1,b=2;
int funct2(int a){
return (b+a);
}
int funct1(int a)
{
    b=funct2(a+1)+1;
    return (b);
}
int main()
{
    int c,a=3;
    for(c=1;c<=5;++c)
    {

        b+=funct1(c+1)+a;
        printf("%d ",b);
    }
    printf("\nAns = %d",b+a);
    return 0;
}
```

"C:\Users\Saem\Deskt...

```
15 43 101 219 457
Ans = 460
Process returned 0 (0x0)
Press any key to continue.
```

2(b):



```c
#include <stdio.h>
int fib(int n)
{
    if(n<=1) return n;
    int ret=fib(n-1)+fib(n-2);
    return ret;
}
int main()
{
    int n=3;
    printf("%d-th Fibonacci number: %d\n",n,fib(n));
    return 0;
}
```

"C:\Users\Saem\Desktop\Solution Final Exam\EzazC223009.exe"

```
3-th Fibonacci number: 2

Process returned 0 (0x0)   execution time : 0.089 s
Press any key to continue.
```

```
                fib(3)
               /      \
          fib(2)        fib(1)
         /     \        /    \
    fib(1)    fib(0)   1      0
    /    \
   1      0
```

1. The initial call is made to fib(3).
2. fib(3) calls fib(2) and fib(1) (the two subproblems).
3. fib(2) further calls fib(1) and fib(0).
4. The base cases are reached when n <= 1. At this point, the recursion stops, and the base cases return the values directly.
5. The recursive calls return their respective values back up the tree.
6. Finally, the value of fib(3) is computed by adding the results of fib(2) and fib(1)

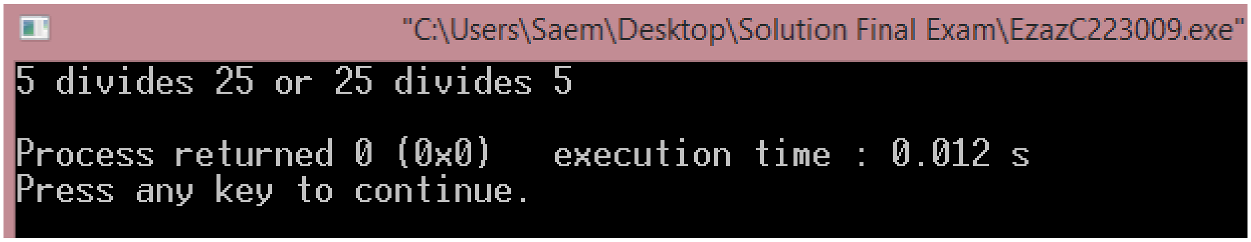
2(b)or:

```
#include <stdio.h>
int F(int n, int k)
{
    if (k == 0 || k == n)
        return 1;
    else
        return F(n − 1, k − 1) + F(n − 1, k);
}

int main()
{
    int n = 3;
    int k = 2;
    int b = F(n, k);
    printf("F(%d, %d) = %d\n", n, k, b);
    return 0;
}
```

2(c):

```
#include <stdio.h>
int divisorcheck(int x, int y)
{
    if (x == 0 && y == 0)
    {
        return 0;
    }
    if (x % y == 0 || y % x == 0)
    {
        return 1;
    }
    return 0;
}
int main()
{
    int x = 5;
    int y = 25;
    if(divisorcheck(x,y))
    {
        printf("%d divides %d or %d divides %d\n", x, y, y, x);
    }
    else
    {
        printf("%d and %d are not divisible\n", x, y);
    }
    return 0;
}
```

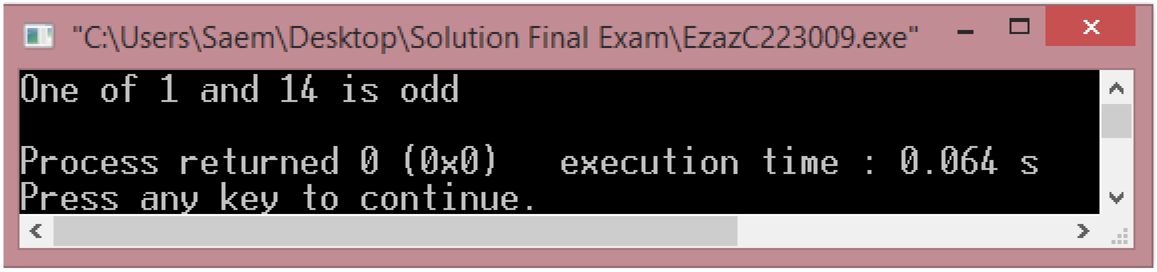


```
5 divides 25 or 25 divides 5

Process returned 0 (0x0)   execution time : 0.012 s
Press any key to continue.
```

2(c) or:

```
#include <stdio.h>
double oddcheck(int x, int y)
{
    if (x % 2 != 0 && y % 2 != 0)
    {
        return 1.1;
    }
    else if (x % 2 != 0 || y % 2 != 0)
    {
        return 0.1;
    }
    else if (x % 2 == 0 && y % 2 == 0)
    {
        return 2.0;
    }
}
int main()
{
    int x = 1;
    int y = 14;
    if (oddcheck(x, y) == 1.1)
    {
        printf("Both %d and %d are odd\n", x, y);
    }
    else if (oddcheck(x, y) == 0.1)
    {
        printf("One of %d and %d is odd\n", x, y);
    }
    else if (oddcheck(x, y) == 2.0)
    {
        printf("Both %d and %d are even\n", x, y);
    }
    return 0;
}
```

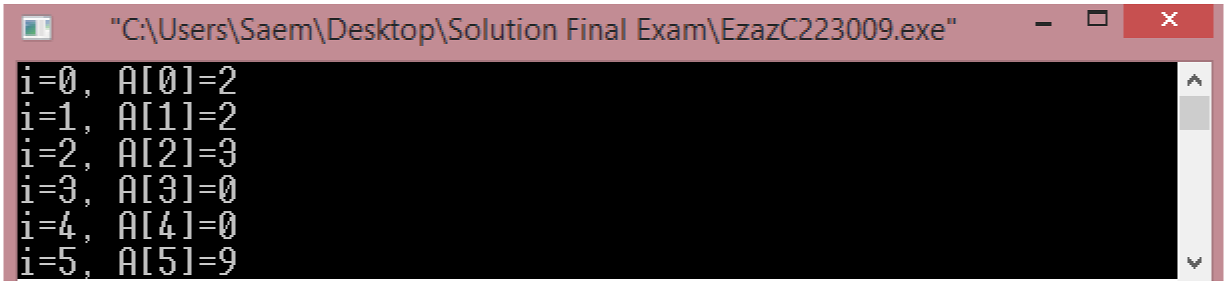


```
One of 1 and 14 is odd

Process returned 0 (0x0)   execution time : 0.064 s
Press any key to continue.
```

3(a)

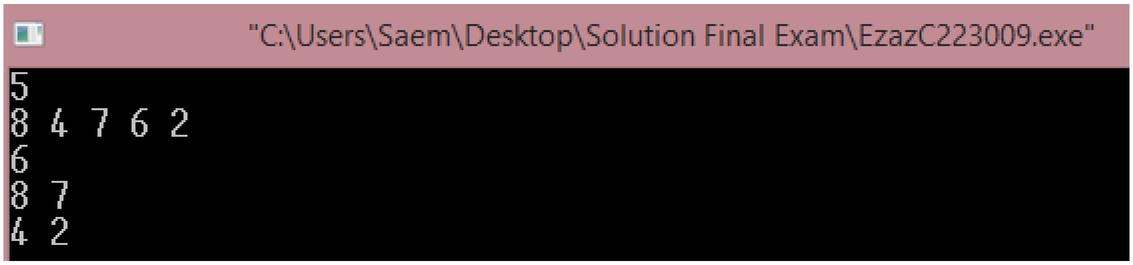
```
#include <stdio.h>
int main()
{
    int a[6] = {2, 2, 3, 0, 0, 9};
    for (int i = 0; i < 6; i++)
    {
        printf("i=%d, A[%d]=%d\n", i, i, a[i]);
    }
    return 0;
}
```



```
i=0, A[0]=2
i=1, A[1]=2
i=2, A[2]=3
i=3, A[3]=0
i=4, A[4]=0
i=5, A[5]=9
```

3(b):

```
#include <stdio.h>
int main()
{
    int N, X;
    scanf("%d", &N);
    int arr[N];
    for (int i = 0; i < N; i++)
    {
        scanf("%d", &arr[i]);
    }
    scanf("%d", &X);
    for (int i = 0; i < N; i++)
    {
        if (arr[i] > X)
        {
            printf("%d ", arr[i]);
        }
    }
    printf("\n");
    for (int i = 0; i < N; i++)
    {
        if (arr[i] < X)
        {
            printf("%d ", arr[i]);
        }
    }
    printf("\n");
    return 0;
}
```

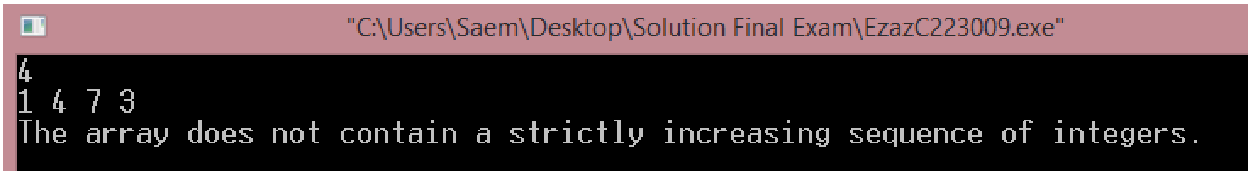


"C:\Users\Saem\Desktop\Solution Final Exam\EzazC223009.exe"

```
5
8 4 7 6 2
6
8 7
4 2
```

3(b) or:

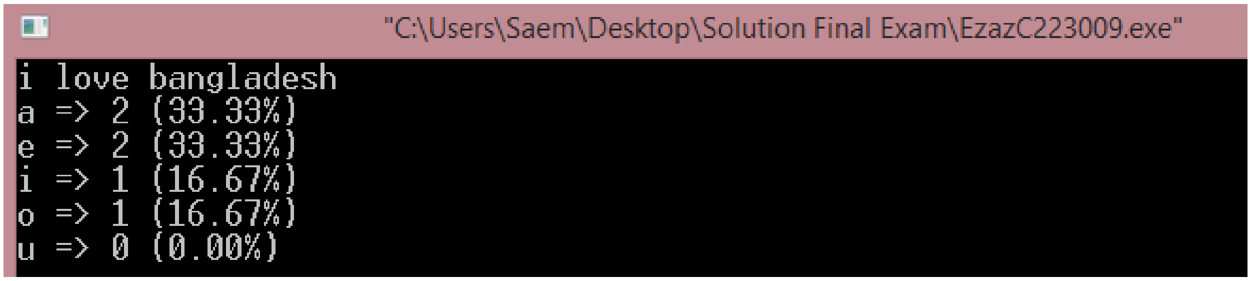
```
#include <stdio.h>
int main()
{
    int a;
    scanf("%d", &a);
    int arr[a];
    for (int i = 0; i < a; i++)
    {
        scanf("%d", &arr[i]);
    }
    int strict = 1;
    for (int i = 1; i < a; i++)
    {
        if (arr[i] <= arr[i - 1])
        {
            strict = 0;
            break;
        }
    }
    if (strict)
    {
        printf("The array contains a strictly increasing sequence of integers.\n");
    }
    else
    {
        printf("The array does not contain a strictly increasing sequence of
integers.\n");
    }
    return 0;
}
```



```
4
1 4 7 3
The array does not contain a strictly increasing sequence of integers.
```

3(c):

```
#include <stdio.h>
#include <string.h>
int main()
{
    char s[1000];
    int vowel_count[5] = {0};
    int total_vowels = 0;
    char vowels[] = {'a', 'e', 'i', 'o', 'u'};
    gets(s);
    for (int i = 0; s[i] != '\0'; i++)
    {
        for (int j = 0; j < 5; j++)
        {
            if (s[i] == vowels[j])
            {
                vowel_count[j]++;
                total_vowels++;
                break;
            }
        }
    }
    for (int i = 0; i < 5; i++)
    {
        printf("%c => %d (%.2f%%)\n", vowels[i], vowel_count[i], (float)vowel_count[i] / total_vowels * 100);
    }
    return 0;
}
```



"C:\Users\Saem\Desktop\Solution Final Exam\EzazC223009.exe"

```
i love bangladesh
a => 2 (33.33%)
e => 2 (33.33%)
i => 1 (16.67%)
o => 1 (16.67%)
u => 0 (0.00%)
```

4(a)

```
#include <stdio.h>
int main()
{
    int items[8] = {3, 7, 9, 2, 1, 4, 0, 5};
    int *ptr = items;
    for (int i = 0; i < 8; i++)
    {
        printf("%d ", *ptr);
        ptr++;
    }
    printf("\n");
    return 0;
}
```

4(b)

The main difference between pass by value and pass by reference is that, in a pass by value, the parameter value copies to another variable while, in a pass by reference, the actual parameter passes to the function.

Example:

Pass by Value:

```
#include <stdio.h>
void swap (int a, int b)
{
    int temp = a;
    a = b;
    b = temp;
}
int main ()
{
    int a = 10;
    int b = 20;
    printf ("Before swap, a = %d, b = %d\n", a, b);
    swap (a, b);
    printf ("After swap, a = %d, b = %d\n", a, b);
    return 0;
}
```

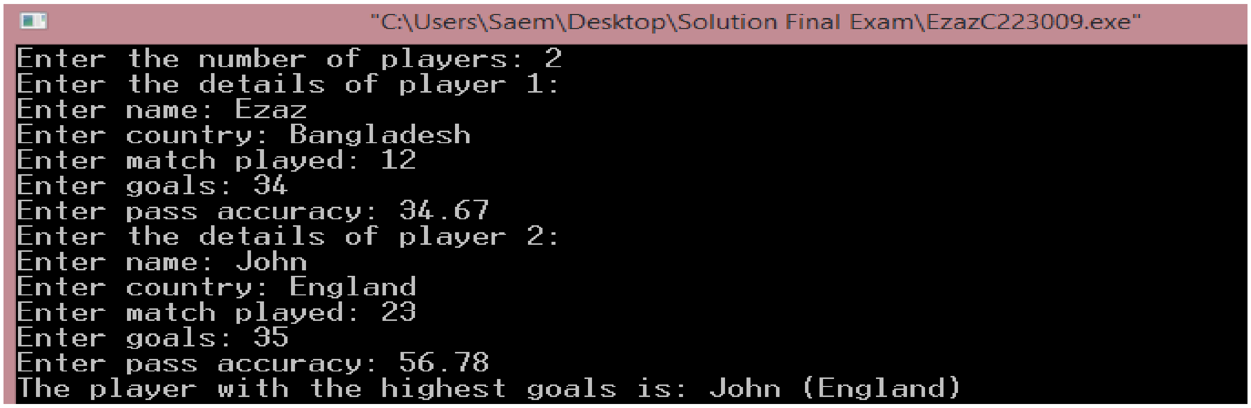
Pass By Reference:

```
#include <stdio.h>
int main ()
{
    int a = 100;
    int b = 200;
    printf("Before swap, value of a : %d\n", a );
    printf("Before swap, value of b : %d\n", b );
    swap(&a, &b);
    printf("After swap, value of a : %d\n", a );
    printf("After swap, value of b : %d\n", b );
    return 0;
}
void swap(int *x, int *y)
{
    int temp;
    temp = *x;
    *x = *y;
    *y = temp;
    return;
}
```

4(c)

```
#include <stdio.h>
struct Player
{
    char name[24];
    char country[24];
    int match_played;
    int goals;
    double pass_accuracy;
};
int main()
{
    int n;
    printf("Enter the number of players: ");
    scanf("%d", &n);
    struct Player Player[n];
    for (int i = 0; i < n; i++)
    {
        printf("Enter the details of player %d:\n", i + 1);
        printf("Enter name: ");
        scanf("%s", Player[i].name);
        printf("Enter country: ");
        scanf("%s", Player[i].country);
        printf("Enter match played: ");
        scanf("%d", &Player[i].match_played);
        printf("Enter goals: ");
        scanf("%d", &Player[i].goals);
        printf("Enter pass accuracy: ");
        scanf("%lf", &Player[i].pass_accuracy);
    }
    int max_goals = 0;
    int max_goals_index = 0;

    for (int i = 0; i < n; i++)
    {
        if (Player[i].goals > max_goals)
        {
            max_goals = Player[i].goals;
            max_goals_index = i;
        }
    }
    printf("The player with the highest goals is: %s (%s)\n",
Player[max_goals_index].name, Player[max_goals_index].country);
    return 0;
}
```



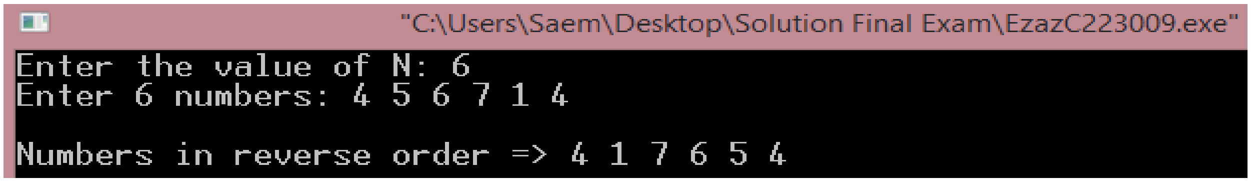
```
Enter the number of players: 2
Enter the details of player 1:
Enter name: Ezaz
Enter country: Bangladesh
Enter match played: 12
Enter goals: 34
Enter pass accuracy: 34.67
Enter the details of player 2:
Enter name: John
Enter country: England
Enter match played: 23
Enter goals: 35
Enter pass accuracy: 56.78
The player with the highest goals is: John (England)
```

5(a):

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main()
{
    char StudentID[100];
    int TotalMarks;
    printf("Enter the student ID: ");
    gets(StudentID);
    printf("Enter the total marks: ");
    scanf("%d", &TotalMarks);
    FILE* file = fopen("db.txt", "a+");
    if (file == NULL)
    {
        printf("Error opening the file.\n");
        return 1;
    }
    FILE* tempFile = fopen("temp.txt", "w");
    if (tempFile == NULL)
    {
        printf("Error creating temporary file.\n");
        fclose(file);
        return 1;
    }
    char existingID[100];
    int marks;
    int found = 0;
    while (fscanf(file, "%s %d", existingID, &marks) == 2)
    {
        if (strcmp(existingID, StudentID) == 0)
        {
            fprintf(tempFile, "%s %d\n", StudentID, TotalMarks);
            found = 1;
        }
        else
        {
            fprintf(tempFile, "%s %d\n", existingID, marks);
        }
    }
    if (!found)
    {
        fprintf(tempFile, "%s %d\n", StudentID, TotalMarks);
    }
    fclose(file);
    fclose(tempFile);
    remove("db.txt");
    rename("temp.txt", "db.txt");
    printf("Student information saved successfully.\n");
    return 0;
}
```

5(a) or:

```
#include<stdio.h>
int main()
{
    FILE *fp = fopen("test.txt", "w");
    int i,n,N;
    printf("Enter the value of N: ");
    scanf("%d", &N);
    printf("Enter %d numbers: ", N);
    for(i = 0; i < N; i++)
    {
        scanf("%d", &n);
        fprintf(fp, "%d ", n);
    }
    fclose(fp);
    printf("\nNumbers in reverse order => ");
    FILE *fpr = fopen("test.txt", "r");
    int a[N];
    for(i = 0; i < N; i++)
    {
        fscanf(fpr, "%d ", &a[i]);
    }
    for(i = N - 1; i >= 0; i--)
    {
        printf("%d ", a[i]);
    }
    printf("\n");
    fclose(fpr);
    return 0;
}
```



```
Enter the value of N: 6
Enter 6 numbers: 4 5 6 7 1 4

Numbers in reverse order => 4 1 7 6 5 4
```

5(c):

Output: 17

Modifying the previous code:

```
#include <stdio.h>
#define PROD(a,b) ((a)*(b))
int main()
{
    printf("%d ", PROD(3+4, 2+6));
    return 0;
}
```

Thanks Everyone Assalamualikum