

2022

x > 1

y = 3 + 4 + 5 + 6 + 7 + 8 - - -

Question #1 (25%)

Given the following series to evaluate y:

y = 3x + 4x³ + 5x⁵ + 6x⁷ + 7x⁹ + ...

(where term 1 = 3x, term 2 = 4x³, term 3 = 5x⁵, term 4 = 6x⁷ and so forth)

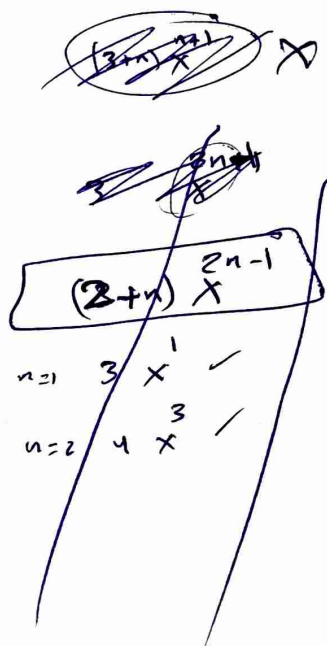
Write a function (only a function) called `getVal` that takes *integer x* as a parameter and uses the above series to calculate and return the number of terms needed before y exceeds (تزيد عن) one million (1000000).

Note: You are NOT allowed to define and use any extra functions or use any of the <math.h> predefined functions (e.g. pow).

```

int getVal ( int x )
{
    int y = 0, n = 1; rem;
    while ( y < 1000000 )
    {
        rem = (2+n) * x2n-1;
        y = y + rem;
        n++;
    }
    return y;
}

```



```

int getVal getVal ( int x )
{
    int y = 0, n = 1, rem;
    while ( y < 1000000 )
    {
        rem = (2+n) * x2n-1;
        y = y + rem;
        n++;
    }
    return n;
}

```

(16)

Question 2 (25%)

A. (15%)

Given the following *if* statement which finds the larger value for any given two different positive integers *n1* and *n2* (assume *n1* and *n2* are always both larger than 100):

```
if ( n1 > n2 )
    printf ( " %d is larger than %d", n1, n2);
else
    printf ( " %d is larger than %d", n2, n1);
```

Rewrite (*inside box*) the above code using only a *switch* statement instead of an *if* statement. Your *switch* should not include any *if* statements anywhere and should be equivalent to and give the same results as the *if* statement above.

```
Switch (n1, n2) {
    case n1 > n2 : printf ( " %d is larger than %d", n1, n2 );
                  break ;
    case
    default : printf ( " %d is larger than %d", n2, n1 );
             break ;
}
```

B. (10%)

Rewrite (*inside box*) the following *for* loop using an equivalent *do/while* loop:

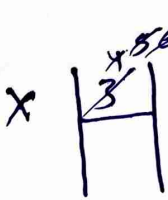
```
for (i=100; i >= 15; i--) // i-- = i minus minus
    printf(" %d \n", (i*2/5));
```

```
do {
    printf ( " %d \n", (i*2/5));
} while (i >= 15) {
    int i = 100;
    i--;
}
```

Question 3 (20%)

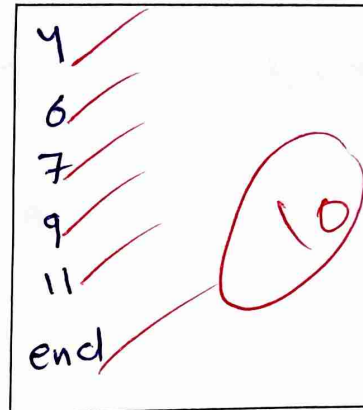
A. (10%)

What is the output of the following loop:



```
int x=3;
while (++x < 12)
{
    printf (" %d \n", x++);
    if (x == 8)
        continue;
    printf (" %d \n", ++x);
}
printf ("end\n");
```

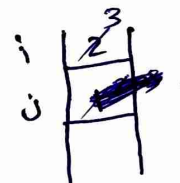
Output



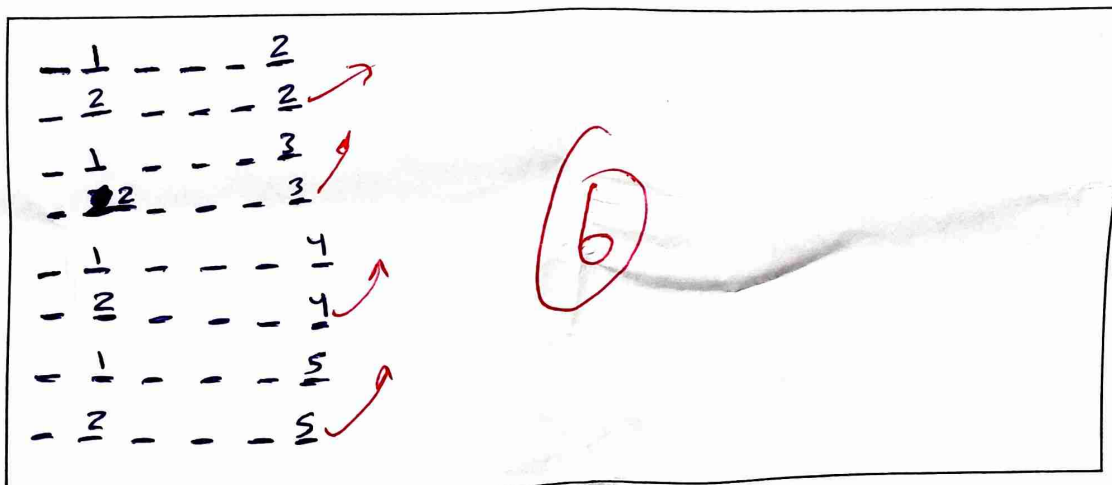
B. (10%)

What is the output of the following loop (*use an underscore (_) for each space*):

```
int i, j;
for ( i=2; i < 6; i++)
{
    for ( j=1; j < 3; j++)
        printf ( "%2d%4d", j, i);
    printf ( "\n");
}
```



Output



16

Question 4 (27%)

Complete the following C program which reads the *length* (integer) and *width* (integer) of a rectangle from a file called *rec.txt* and prints the *area* ($area = length * width$) and *circumference* ($circumference = 2 * length + 2 * width$) of the rectangle to the screen.

The program MUST only use one *void* function called *rectangleInfo* that takes exactly two parameters (arguments) (values of *length* and *width*) and calculates and returns both the *area* and *circumference* of the rectangle to *main*.

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

```
void rectangleInfo ( int *length , int *width );
```

```
int main()
```

```
{
int length, width; FILE *in;
```

```
int a, c ;
```

```
// code to open file rec.txt and read length and width
```

```
in = fopen ("rec.txt", "r");
```

```
fscanf (in, "%d %d", &length, &width);
```

```
// code to call function rectangleInfo and get the area and circumference
// and then print them to screen
```

```
a = area (&length, &width);
```

```
c = circumference (&length, &width);
```

```
printf("Area = %d Circumf = %d", a, c);
```

```
fclose(in);
```

```
return 0; }
```

```
void rectangleInfo ( int *length , int *width )
```

```
{
// fill the code for the function body below
```

```
int *area, *circumference ;
```

```
area = length * width ;
```

```
circumference = (2 * length) + (2 * width) ;
```

```
}
```