Câu 1.

a) Try (int i)

Begin

B1. for < tất cả các giá trị V có thể gán cho A[i] >

A[i] = V

Tong = tong + A[i];

B2. if (i = 3 , tong < 15)

thông báo cấu hình tìm được ra màn hình

B3 else

thử cấu hình tiếp theo

Try (i + 1)

~~B4~~

B4: xoá cấu hình đã chọn ra ngoài

Tong = Tong - A[i] ;

end if

End

b)
```
Void Try ( int i )
{
  for ( int j = 0 ; j <= 9 ; j++ )
  { ~~if ( i > 1 )~~
      if ( i == 1 && j == 0 ) ~~break;~~ Continue;

      A[i] = j;
      Tong = Tong + A[i];
      if ( i == 3 && Tong < 15 )
          Out ();
      else Try ( i + 1 );

      Tong = Tong - A[i];
  }
}
```

Với hàm Out() là hàm xuất cấu hình ra ngoài màn hình

Câu 2:

```
void ThamLam(int A[], int B[], int n, int k)
{
    int X[n];      int dem = 0;
    for (int i = 0; i < n; i++)
        if (A[i] % 2 == 0)
        {
            X[dem] = A[i]; dem++;
        }
    for (int i = 0; i < dem; i++)
        for (int j = i+1; j < dem; j++)
            if (X[i] < X[j])
            {   int tg = X[i];
                X[i] = X[j];
                X[j] = tg;
            }
    if (k > dem) printf(" Khong du so phan tu chan trong A.");
    else
        for (int i = 0; i < k; i++)
            B[i] = X[i];
}
```

Câu 3:

a)

```
int Chiadetri ( int A[], int x, int y)
{
    if (y - x <= 1) return Min ( A[x], A[y] );
    else {
    int min1 , min2;
    min1 = Chiadetri ( A, x, (x+y)/2 );
    min2 = Chiadetri ( A, (x+y)/2 +1, y );
    return Min ( min1, min2 );
    }
}
```

với hàm Min(x, y) là hàm trả về giá trị nhỏ nhất giữa x và y.

b) Gọi T(n) là độ phức tạp của thuật toán

$n = y - x + 1 > 0$

Ta có:

$$T(n) \begin{cases} = 1 & \text{nếu } n = 1 \\ = 2T(n/2) + 2 & \text{nếu } n > 1 \end{cases}$$

$$\begin{aligned} T(n) &= 2T(n/2) + 2 \\ &= 2^2 T(n/4) + 2^2 + 2 \\ &= 2^3 T(n/8) + 2^3 + 2^2 + 2 \\ &= \dots \\ &= 2^i T(n/2^i) + \sum_{k=1}^{i} 2^k \end{aligned}$$

Giả sử n có dạng: $n = 2^k$

$$\Rightarrow T(n) = 2^{\log n - 1} T(2) + \sum_{k=1}^{\log n - 1} 2^k$$

$$= 2^{\log n - 1} + \frac{1 - 2^{\log n - 1 + 1}}{1 - 2} = \frac{2^{\log n}}{2} + 2^{\log n} - 1$$

$$= \frac{3}{2} n - 1$$

Vậy, độ phức tạp của thuật toán: O(n)