//Eric Goulart da Cunha - 2110878
//João Pedro Biscaia Fernandes - 2110361

Questão 1:

O vetor a é alocado na memória compartilhada com 10.000 posições para inteiros.
O vetor soma é alocado na memória compartilhada com 10 posições para inteiros, uma para cada processo trabalhado.

O vetor a é inicializado com o valor 5 em todas as posições.
O vetor soma é inicializado com 0 em todas as posições.

O programa cria 10 processos filhos usando fork().
Cada processo filho trabalha em uma parcela diferente do vetor **a** (1.000 posições por processo), multiplicando os valores por 2 e somando os resultados em soma[i].

Após a conclusão dos processos filhos, o processo pai aguarda cada um deles com waitpid().
O processo pai então soma os resultados parciais de soma para obter a soma total das posições processadas.

O programa desassocia (shmdt) e remove (shmctl) os segmentos de memória compartilhada.

Questão 2:

Da mesma forma que na questão 1, alocamos 10000 posições para o vetor a, sendo a única alocação necessária no programa. Inicializamos de novo o vetor com 5 em cada posição, e criamos um loop para criar os 2 filhos e a cada iteração verificar todos os números do vetor.

Para isso, criamos uma variável verificador, com valor inicial de 5, sendo este o valor esperado inicial para o vetor, e a cada iteração multiplicamos o valor contido na posição do vetor por 2 e posteriormente somamos 2.(Feito da seguinte forma):

```
while (i < 2){
  if((id = fork()) < 0){
    puts ("Erro na criação do novo processo");
    exit (-2);
  }else if(id == 0){
    for(int j = 0; j < 10000; j++){
      temp = a[j]*2+2;
      a[j] = temp;
    }
    exit(0);
```

A verificação ocorre vendo se algum item do vetor difere do verificador, implicando que estão sendo realizados processos ao mesmo tempo alterando os valores do vetor. foi formulado da seguinte forma:

```
}else{
    for(int l = 0; l < 10000; l++){
      if(a[l] != verificador){
        printf("Número diferente: %d -- verificador = %d índice: %d Iteração %d\n", a[l],verificador, l, i+1);
        verificador = a[l];
      }
    }
    waitpid(-1, &status, 0);
}
```

(Esse "else" corresponde ao processo pai e à sua verificação.)
Resulta na seguinte resposta no terminal, em alguns casos.

```
Número diferente: 12 -- verificador = 5 índice: 0 Iteração 1
Número diferente: 12 -- verificador = 26 índice: 9216 Iteração 1
```

Nota-se que o índice e os valores mudam a cada vez que é rodado o programa, justamente devido ao fato de existir essa concorrência.

No fim do programa, liberamos a memória compartilhada da mesma forma que no exercício 1.