Questao 01.

1. Um objeto de massa m é solto de uma altura h_0 em relação ao solo. Após t segundos a sua altura é dado pela expressão:
$h(t) = h_0 - \frac{mg}{k}t + \frac{m^2 g}{k^2}(1 - e^{\frac{-kt}{m}})$
onde k é o coeficiente de resistência do ar e g a aceleração da gravidade. Sendo $m = 1kg$, $h_0 = 30m$, $k = 0.5kg/s$ e $g = 9.8m/s^2$, estime o tempo que o objeto leva para chegar ao solo utilizando o método da Falsa Posição, com precisão de 0.001 e máximo de 5 iterações.

$$h(x) = 30 - \frac{9{,}8x}{0{,}5} + \frac{9{,}8}{0{,}25} \cdot (1 - e^{-0{,}5x})$$

$$h(3) = 1{,}65329 \qquad h(4) = -14{,}5051$$

Neste caso, pelo teorema do valor intermediário, há uma raiz entre 3 e 4.

$$[3;4] \rightsquigarrow \underline{x} = \frac{af(b) - bf(a)}{f(b) - f(a)}$$

$$\underline{x} = \frac{3 \cdot -14{,}50510 - (4 \cdot 1{,}65329)}{-14{,}50510 - 1{,}65329}$$

$$\underline{x} = 3{,}10232$$

$$f(\underline{x}) = 0{,}0840854$$

Como F(x̄) * F(b) < 0, usaremos novamente o teorema do valor intermediário e trocaremos o a.Como |b-a| é maior que a precisao e |F(x̄)| também é menor que a precisao, continuaremos testando. Portanto [3,10232 ; 4].

$f(3{,}10232) = 0{,}0840854 \quad f(4) = -14{,}5051$

$$\underline{x} = \frac{3{,}10232 \cdot (-14{,}5051) - 4 \cdot 0{,}0840854}{-14{,}5051 - 0{,}0840854}$$

$\underline{x} = 3{,}10749 \quad f(\underline{x}) = 0{,}00414931$

Como F(x̄) * F(b) < 0, usaremos novamente o teorema do valor intermediário e trocaremos o a.Como |b-a| é maior que a precisao e |F(x̄)| também é menor que a precisao, continuaremos testando. Portanto [3,10749 ; 4].

$f(3{,}10749) = 0{,}00414931$

$f(4) = -14{,}5051$

$$\underline{x} = \frac{3{,}10749 \cdot (-14{,}5051) - 4 \cdot 0{,}00414931}{-14{,}5051 - 0{,}00414931}$$

$\underline{x} = 3{,}10775 \quad f(\underline{x}) = 0{,}000204445$

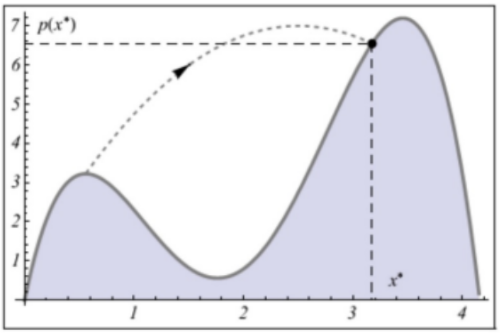
Como |F(x̄)| < e pararemos a execucao, pois a precisao foi alcancada. Portanto levará, aproximadamente, 3,10775 segundos para o objeto chegar ao solo.
Para a solucao desta questao eu fiz uso deste código C++ que eu mesmo implementei.

```cpp
double f(double x) {
    const float EULER = 2.718281;
    return 30 - ((9.8 / 0.5) * x) + ((9.8 / 0.25) * (1 - pow(EULER, (x* -0.5))));
}

void questao(double a, double b, int iterator, double precision) {
    for (int i = 0; i < iterator; i++) {
        cout << "[" << a << ";" << b << "]" << endl;
        double resA = f(a);
        double resB = f(b);
        cout << "F(a) = " << resA << " F(b) = " << resB << endl;
        double falsePosition = ((a * resB) - (b * resA)) / (resB - resA);
        double resFalse = f(falsePosition);
        cout << "FalsaPosicao = " << falsePosition << " F(FalsaPosicao) = " << resFalse << endl;
        if (resA * resFalse < 0) {
            b = falsePosition;
        } else if (resB * resFalse < 0) {
            a = falsePosition;
        } else {
            cout << "Vixi kk" << endl;
            break;
        }
        cout << "Iteração:" << i+1 << endl;
        cout << "Novos A e B" << endl;
        cout << "A = " << a << " B = " << b << endl << endl << endl;
        if (abs(b-a) < precision || abs(resFalse) < precision) {
            cout << "Precisão alcançada. Parando iteração." << endl;
            break;
        }
    }
}
```

Questao 02.

2. É preciso lançar um projétil de um morro para outro onde estão as forças inimigas. As dois morros são representado pela região sombreada do gráfico da função $p(x) = -x^4 + 7.7x^3 - 18x^2 + 13.6x$, como mostra a figura abaixo. O projétil é lançado do moro de menor elevação para o de maior e sua curva é descrita pela função $q(x) = -x^2 + 5x + 0.75$. Qual a altura que irá ocorrer o impacto com a maior elevação? Utilize o método de Falsa Posição com precisão 0.001 e máximo de 5 iterações.



Comecei pensando em como trazer esse ponto de interseccao para a origem. Entao fiz f(x) = p(x) - q(x) e use o GeoGebra para ver o comportamento.

Entao percebi que agora bastava encontrar a raiz que esta entre 3 e 3,5 utilizando minha nova f(x). Pontanto inicei em [3;3,5].

$f(3) = -1,05$ $\quad$ $f(3.5) = 1,175$

$\underline{x} = 3,23596$ $\quad$ $f(\underline{x}) = 0.329473$

Verifiquei se |a-b| < e ou |f(x̄)| < e. Neste caso, ainda nao é verdade. Entao como f(a)*f(x̄) < 0, substitui b por x̄. Portanto [3;3,23596].

$f(3) = -1,05$ $\quad$ $f(3,23596) = 0.329473$

$\underline{x} = 3,1796$ $\quad$ $f(\underline{x}) = 0.0366111$

Verifiquei se |a-b| < e ou |f(x̄)| < e. Neste caso, ainda nao é verdade. Entao como f(a)*f(x̄) < 0, substitui b por x̄. Portanto [3;3,1796].

$f(3) = -1,05$ $f(3,1796) = 0,0366111$

$\underline{x} = 3,17355$ $f(\underline{x}) = 0,00347829$

Verifiquei se |a-b| < e ou |f(x̄)| < e. Neste caso, ainda nao é verdade. Entao como f(a)*f(x̄) < 0, substitui b por x̄. Portanto [3;3,17355].

$f(3) = -1,05$ $f(3,17355) = 0,00347829$

$\underline{x} = 3,17298$ $f(\underline{x}) = 0,000325112$

Agora |f(x̄)| < e, portanto, a precisao foi alcancada quando x̄ = 3,17298. Mas este resultado é suficiente para a f(x), a questao busca saber a altura do impacto com maior elevecao. Logo, basta usar x̄ na funcao q(x) que descreve a trajetória do projetio. Q(x̄) = 6.5471. Entao essa a altura do impacto com maior elevacao.

Para resolver essa questao eu também fiz uso de um algoritmo em C++ implementado por mim.

```cpp
double p(double x) {
    return -pow(x, 4) + 7.7 * pow(x, 3) - 18 * pow(x, 2) + 13.6 * x;
}

double q(double x) {
    return -pow(x, 2) + 5 * x + 0.75;
}

double f(double x) {
    return p(x) - q(x); //Retorna a nova funcao
}
```

```cpp
double falsa_posicao(double a, double b, int iterator, double precision) {
  double resA,resB,resFalse, falsePosition;

    for(int i = 0; i < iterator; i++) {
        cout << "[" << a << ";" << b << "]" << endl;
        resA = f(a);
        resB = f(b);
        cout << "F(a) = " << resA << " F(b) = " << resB << endl;
        falsePosition = ((a * resB) - (b * resA)) / (resB - resA);
        resFalse = f(falsePosition);
        cout << "FalsaPosicao = " << falsePosition << " F(FalsaPosicao) = " << resFalse << endl;
        if (resA * resFalse < 0) {
            b = falsePosition;
        } else if (resB * resFalse < 0) {
            a = falsePosition;
        } else {
            cout << "Vixi kk" << endl;
            break;
        }
        cout << "Iteração:" << i+1 << endl;
        cout << "Novos A e B" << endl;
        cout << "A = " << a << " B = " << b << endl << endl << endl;
        if (abs(b-a) < precision || abs(resFalse) < precision) {
            cout << "Precisão alcançada. Parando iteração." << endl;
            break;
        }
    }
    return falsePosition;
}
```