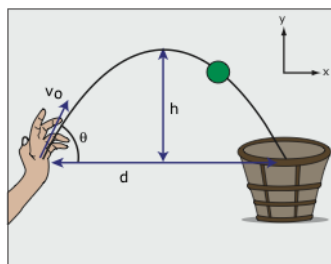


Programação 2024/2025 – 2º Semestre**Aula de Laboratório 2**

Ao desenvolver os seguintes programas tenha em atenção o bom uso dos comentários, o uso da indentação e o correto nome das variáveis!!!! Bom trabalho!

1. Escreva um programa que receba o número de total de segundos como um inteiro e imprima o número total de horas, minutos e segundos correspondente. Não utilize quaisquer condições ou ciclos.
2. Escreva um programa que leia dois tempos, cada um especificado em horas (0-23), minutos (0-59) e (0-59) segundos; sendo o primeiro tempo maior do que o segundo, determine a diferença entre eles e escreva-a, também, em horas, minutos e segundos.
3. Os projéteis seguem uma trajetória parabólica no ar devido à aceleração da gravidade. Considere a equação em baixo que descreve a relação entre o ângulo de partida θ , velocidade inicial v_0 e distância d , onde $g = 9.8 \text{ m/s}^2$. De forma a que o projétil não bata no teto também é necessário verificar que a altura não excede um valor especificado pelo utilizador. Assuma que o projétil é enviado do chão (altura = 0). Escreva um programa que receba como parâmetros o ângulo de partida θ em graus, velocidade inicial v_0 e altura do teto, indica a que distância deve estar o centro do cesto e imprime uma mensagem a indicar se o projétil bate ou não no teto. Note que a função sin do C recebe o ângulo em radianos.

$$d = \frac{v_0^2 \sin(2\theta)}{g}$$
$$h = \frac{v_0^2 \sin^2(\theta)}{2g}$$



4. Escreva um programa para ler, separadamente, três caracteres correspondentes a dígitos decimais (algarismos) e escreva o dobro do número inteiro cujos algarismos são os caracteres lidos.
5. A radiação eletromagnética pode ser classificada em uma de sete categorias de acordo com a sua frequência, tal como indicado na tabela em baixo. Escreva um programa que receba a frequência e imprime o nome da classe correspondente. Use a directiva #define recebendo a frequência como parâmetro.

Name	Frequency range (Hz)
Radio waves	Less than 3×10^9
Microwaves	3×10^9 to less than 3×10^{12}
Infrared light	3×10^{12} to less than 4.3×10^{14}
Visible light	4.3×10^{14} to less than 7.5×10^{14}
Ultraviolet light	7.5×10^{14} to less than 3×10^{17}
X-rays	3×10^{17} to less than 3×10^{19}
Gamma rays	3×10^{19} or more

6. Escreva um programa para ler o número de um mês e que imprima de seguida o número de dias para este mês. Utilize o switch/case. Considere um ano não bissexto.

7. Escreva um programa para ler um número real positivo e que realize o arredondamento para a n -ésima casa decimal, com n introduzido pelo utilizador (n assume valores entre 1 e 7). O valor deve ficar guardado arredondado numa variável. Realize o programa sem recurso às funções de biblioteca do C e utilize o printf para produzir o mesmo valor.
Exemplo: para os valores de 2.163, 0.547 e 0.997, e supondo que se especifica $n = 2$, o programa deve imprimir, respetivamente 2.16, 0.55 e 1.00.

8. Considere o problema 3 e que o cesto tem um diâmetro c . Escreva um programa que calcule uma gama de valores de ângulos aceitáveis de forma que o projétil caia dentro do cesto para uma dada distância e velocidade inicial. Assim, o programa deve ler como parâmetros de entrada a distância d , a velocidade inicial v_o , o diâmetro do cesto c e imprimir a gama de ângulos (em graus). Utilize um ciclo *while* e não derive novas equações para resolver este problema.

Anexo – Uso de bibliotecas

Para alguns problemas é necessário usar funções matemáticas, por exemplo *sin*. Estas funções estão declaradas no ficheiro `math.h` e necessitam da biblioteca matemática. Assim é necessário escrever `#include<math.h>` no início do ficheiro com o código C e configurar o Code::Blocks para usar a biblioteca matemática.

De modo a perceber o funcionamento de uma função (*sin*, *cos*, *mod*, *ceil*, por exemplo) pode-se aceder ao manual de cada uma delas. Abra um terminal:

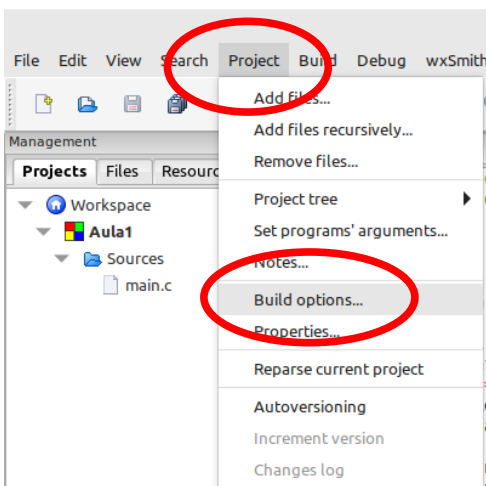


Dê um dos comandos no terminal:

- `man sin`
- `man cos`
- `man mod`
- `man ceil`

Dependendo da aplicação poderá ser necessário usar bibliotecas externas: biblioteca matemática, biblioteca gráfica, etc. Assim, para usar essas bibliotecas, é necessário compilar dando a conhecer ao gcc a sua existência. Caso use a linha de comandos para usar a biblioteca matemática é necessário usar a opção `-lm` do compilador gcc. No entanto, o *Code::Blocks* facilita a vida do programador, ao permitir-lhe indicar uma única vez quais as bibliotecas que a aplicação necessita. A partir desse momento, basta ao programador compilar sem se preocupar com os argumentos do gcc.

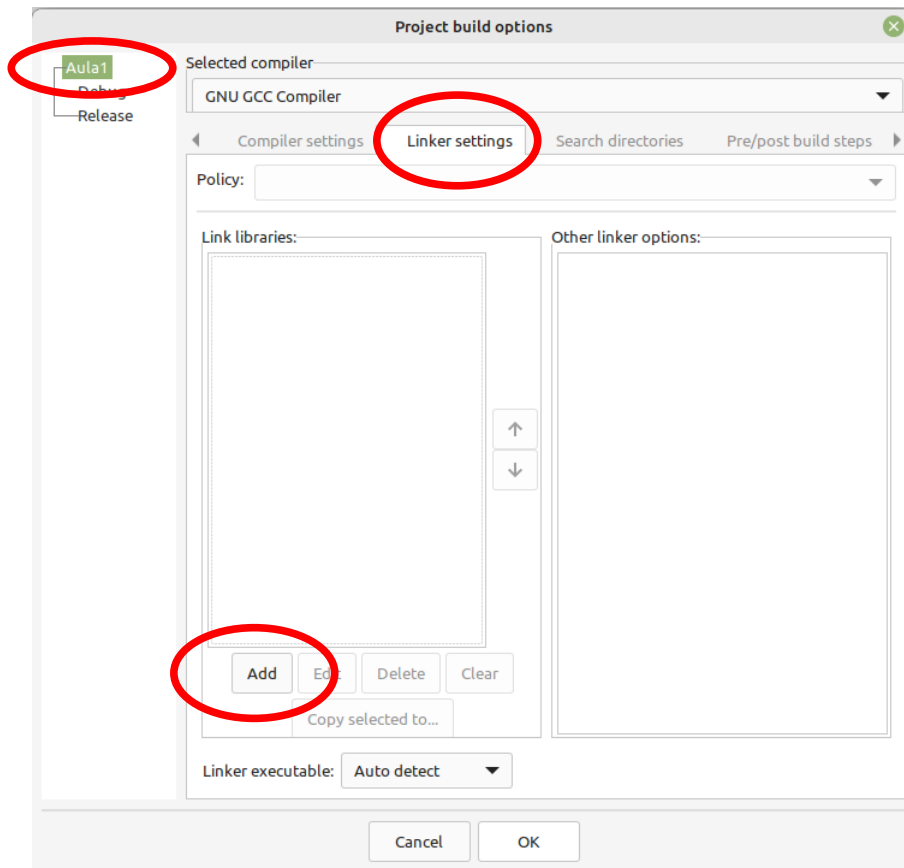
Para adicionar uma biblioteca a um projecto no Code::Blocks, seleccionar o menu Project → *Build options*:



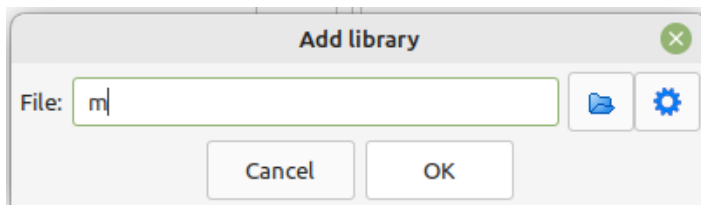
Na nova janela, seleccionar a patilha *Linker Settings*.

No lado esquerdo da janela seleccionar o nome do projecto, caso não esteja já seleccionado.

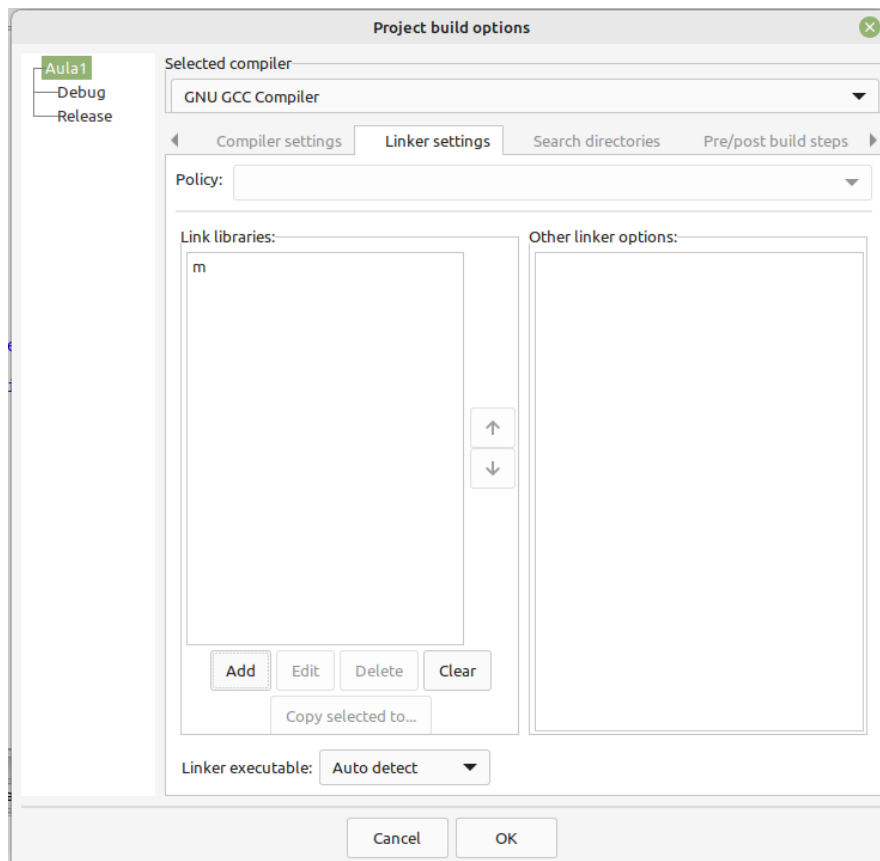
Carregar no botão *Add* para adicionar a biblioteca.



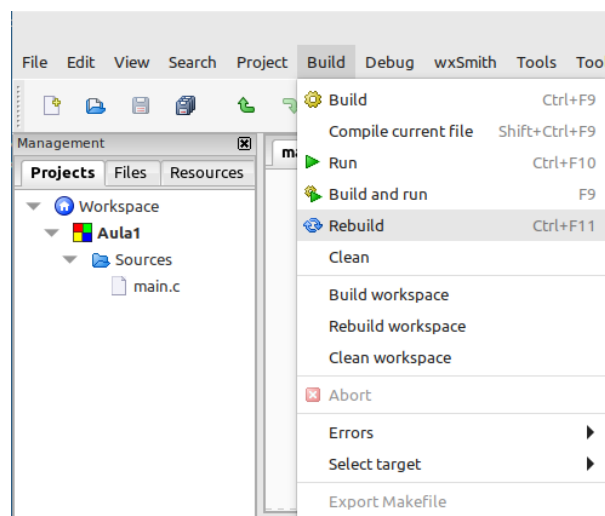
Para cada biblioteca deverá ser adicionado o seu nome. Adicionar a biblioteca *math* (“m”) e carregar **OK**:



Atenção que apenas deve introduzir o nome da biblioteca. Após todas as bibliotecas terem sido adicionadas deve-se carregar no botão **OK**. Neste caso, apenas se introduz uma biblioteca.



Deverá ser necessário escolher a opção *Build* ou *Rebuild* para compilar o projecto:



Depois é apenas necessário executar o projecto com a opção *Run*.