

# Universidad Nacional de Asunción - Facultad Politécnica

## Primer Examen Final de Programación

Fecha: 7 de noviembre de 2020 – Grupo 1

### Directivas y Recomendaciones

- Los problemas deben de ser resueltos usando el Lenguaje C y las herramientas vistas hasta la fecha del examen.
- Los comentarios aclaratorios pueden ayudar en la corrección del examen, influyendo favorablemente en su calificación.
- Utilice en sus programas los nombres de las estructuras indicadas en el enunciado.
- La duración del examen es de 180 minutos.
- Los programas en código fuente deben subirse al aula virtual de la materia (en EDUCA).
- La interpretación del problema es parte de la evaluación del examen
- Solo los ejercicios que compilan tendrán puntos y los mismos serán evaluados por casos de pruebas establecidos por el Profesor.
- El profesor estará evaluando los ejercicios con alguna herramienta de comparación de códigos. Códigos iguales o muy semejantes no serán considerados para la corrección

### Ejercicio 1 (50 Pts)

La matriz denominada matriz Hadamard  $H(N)$  es una matriz cuadrada de tamaño  $N \times N$  de valores lógicos con una propiedad interesante que consiste en que cualquiera de dos filas de la matriz difiere en exactamente  $N/2$  valores. Esta última propiedad la hace útil para el diseño de códigos de detección de error.  $H(1)$  es una matriz  $1 \times 1$  con un solo elemento cuyo valor es verdadero, y para cualquier  $N > 1$ ,  $H(2N)$  es obtenida alineando cuatro copias de  $H(N)$  en un cuadrado mayor ( $2N \times 2N$ ), e invirtiendo todos los elementos (el que era falso a verdadero y viceversa) en el cuadrado  $N \times N$  ubicado abajo a la derecha. A continuación, se muestran 4 ejemplos de matrices Hadamard (en negrita el bloque de cada matriz que cambia de sentido):

H (1)	H (2)		H (4)				H (8)							
V	V	V	V	V	V	V	V	V	V	V	V	V	V	
V	V	F	V	F	V	F	V	V	F	V	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	F	F	V	V	F	F	V	V	F	F	V	V	F	
V	F	F	V	F	F	V	V	F	F	V	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F	F	V	F	
V	V	F	F	V	V	F	F	V	V	F				

Se pide que construya una función que reciba un número  $N$  y retorne un puntero  $*p$  de manera a acceder a un arreglo para poder imprimir la matriz de Hadamard  $H(N)$  en la función principal (main). El parámetro  $N$  pasado a la función debe ser un número potencia de 2 ( $2^x == N$ ), si no es así el puntero  $*p$  debe de apuntar a una matriz con puros valores 0 (ceros) y un mensaje indicando que  $N$  no es potencia de 2. La función debe funcionar para  $N \leq 2048$ . Validar que  $1 \leq N \leq 2048$ , si no es así, indicar este error y una matriz con puros valores 0 (ceros).

Observación:

- Las validaciones deben quedar dentro de la función.

**Entrada del Programa:** Valor  $N$  introducido por usuario.

**Salida del Programa:** Impresión de la matriz de Hadamard  $H(N)$

### Valoración del Programa (Puntaje):

Respetar las entradas y las salidas esperadas. (25 % de ejercicio)

La solución de la función es correcta y respeta el parámetro de entrada esperado y salida esperada. (25 % de ejercicio)

Utiliza aritmética punteros dentro de la función programada (25 % de ejercicio)

La salida del programa es correcta (25 % de ejercicio)

## Ejercicio 2 (50 Pts)

Escriba un programa en C que dado un archivo (numeros.txt) cuyo contenido tiene números separados por espacio, tabulador o salto de línea pueda generar otros tres archivos (pares.txt, impares.txt y primos.txt) con los números pares, impares y primos en archivos separados y obtenidos de numeros.txt. En el caso que sea primo solo debe de aparecer en el archivo primos.txt.

A continuación, se muestra un ejemplo de archivo numeros.txt

```
73771782 81296771 79982326 75332246 10128193
81643413 76259734 94432076 50063976 91748657
42311916 -1920042 90747362 53851612 43498487
73193311 96685173 39019033 8630045 59322952
```

**Las salidas esperadas son:**

Para el archivo pares.txt

```
73771782
79982326
75332246
76259734
94432076
50063976
42311916
-1920042
90747362
53851612
59322952
59322952
```

Para el archivo impares.txt

```
81296771
81643413
91748657
43498487
96685173
8630045
```

Para el archivo primos.txt

```
10128193
73193311
39019033
```

Observación:

- El archivo numeros.txt debe de leerse del programa principal, sin la intervención del usuario.

**Valoración del Programa (Puntaje):**

Lee correctamente el archivo numero.txt en el programa principal (25 % de ejercicio)

Escribe correctamente el archivo pares.txt (25 % de ejercicio)

Escribe correctamente el archivo impares.txt (25 % de ejercicio)

Escribe correctamente el archivo primos.txt (25 % de ejercicio)