

Escuela de Verano GPU2014

Examen 1

June 11, 2014

1. A continuación se presenta un programa, que se puede compilar. ¿ Qué arroja cuando se ejecuta ?
 - (a) Result = 20
 - (b) Result = 14
 - (c) Result = 28
 - (d) Result = 2
 - (e) Result = 4
 - (f) Segmentation fault

```

#include <stdio.h>

double transform1 (double input)
{
    return 2.0*input;
}

void transform2 (double input)
{
    input = 3.0*input;
}

void transform3 (double *input)
{
    *input *= 4.0;
}

int main () {
    double x = -2.0;
    double y = +4.0;
    transform2 (x);
    transform3 (&y);
    printf ("Result = %g\n", transform1 (x+y));
}

/* Cual es la salida de este programa ?

1. Result = 20
2. Result = 14
3. Result = 28
4. Result = 2
5. Result = 4
6. Segmentation fault

*/

```

2. A continuación se presentan 3 breves programas (por orden de aparición `test1.c`, `test2.c` y `test3.c`) que pretenden hacer lo siguiente:

- (a) Crear tres arreglos (vectores) de 10 elementos cada uno, llamados `x`, `y` y `z`.
- (b) Inicializar los valores de `x` e `y`.
- (c) Multiplicar estos dos vectores elemento por elemento y poner los 10 resultados en el vector `z`.
- (d) Escribir los resultados en pantalla.

Contestar las siguientes preguntas:

- (a) ¿ Cual de estos 3 programas va a imprimir los resultados ?
- (b) ¿ Cual va a arrojar, además de los resultados, un mensaje de error ?
- (c) ¿ Cual no va a arrojar ningún resultado (y a lo mejor no va a compilar) ?
- (d) ¿ Que mejoría propondría al programa que arroja los resultados sin mensaje de error ?

```

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#define NB 10

int main () {
    int i;
    //Reservacion de memoria
    float x[NB], y[NB], z[NB];

    //Inicializacion de los arreglos
    for (i = 0; i < NB; i++) {
        x[i] = (float)i*0.34;
        *(y+i) = pow((float)i,1.3);
    }

    //Calculo del producto
    for (i = 0; i < NB; i++)
        z[i] = x[i] * y[i];

    //Impresion del resultado
    for (i = 0; i < NB; i++) {
        printf ("%f * %f = %f\n",x[i], y[i], z[i]);
    }

    free(x);
    free(y);
    free(z);
}

```

```

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#define NB 10

int main () {
    int i;
    //Reservacion de memoria
    float *x, *y, *z;
    x = malloc (sizeof(float) * NB);
    y = malloc (sizeof(float) * NB);
    z = malloc (sizeof(float) * NB);

    //Inicializacion de los arreglos
    for (i = 0; i < NB; i++) {
        x[i] = (float)i*0.34;
        *(y+i) = pow((float)i,1.3);
    }

    //Calculo del producto
    for (i = 0; i < NB; i++) {
        z[i] = x[i] * y[i];
    }

    //Impresion del resultado
    for (i = 0; i < NB; i++) {
        printf ("%f * %f = %f\n",x[i], y[i], z[i]);
    }
}

```

```

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#define NB 10

int main () {
    int i;
    //Reservacion de memoria
    float *x, *y, *z;
    x = malloc (sizeof(float) * NB);
    y = malloc (sizeof(float) * NB);
    z = malloc (sizeof(float) * NB);

    //Inicializacion de los arreglos
    for (i = 0; i < NB; i++) {
        x[i] = (float)i*0.34;
        x[i] = pow((float)i,1.3);
    }

    //Calculo del producto
    z = x * y;

    //Impresion del resultado
    for (i = 0; i < NB; i++) {
        printf ("%f * %f = %f\n",x[i], y[i], z[i]);
    }

    free(x);
    free(y);
    free(z);
}

```