

Indicaciones específicas:

Duración máxima: 60 minutos

Número de preguntas: 5

Se permite el uso de copias, apuntes, libros y toda información necesaria.

Pregunta 1 (5 puntos)

Para la calculadora que se provee implementar las siguientes operaciones con un nuevo tipo de datos "string" que contenga únicamente caracteres de letras mayúsculas o minúsculas:

`~string1`

proporciona una cadena de caracteres con los caracteres intercambiados de minúsculas a mayúsculas y viceversa. Esta operación unitaria debe tener la prioridad mas alta entre las operaciones.

`string1 && string2`

proporciona una cadena concatenada de `string1` y `string2` con todos los caracteres convertidos a letras minúsculas.

`string1 ## string2`

proporciona una cadena donde se suprimen las ocurrencias de `string2` dentro de `string1`. Esta operación tiene precedencia de evaluación de derecha a izquierda.

`&&` debe tener prioridad sobre `##`.

Ejemplos:

`aabbccbbdde ## bb = aaccddee`

`AABB && ~cc ## bbcc ## cc = aacc`

Pregunta 2 (5 puntos)

Para el siguiente código en lenguaje C:

```
typedef struct complejo {
    double re;
    double im;
} Complejo;

Complejo sumaComplejos(Complejo a, Complejo b)
{
    Complejo c;
    c.re = a.re + b.re;
    c.im = a.im + b.im;
    A: return c;
}

int main()
{
    Complejo c1 = {1,1}, c2 = {2,2}, c3;

    c3 = sumaComplejos(c1, c2);
    printf("%f + %fi", c3.re, c3.im);
}
```

```
    return 0;
}
```

- Con respecto al paso de parámetros haciendo uso de la pila. ¿Es por valor o por referencia? Explique como se realiza el paso de parámetros en este caso para la función sumaComplejos.
- Explique cómo la función sumaComplejos realiza el retorno de su valor.
- Explique cómo la función main hace uso del valor retornado por la función sumaComplejos.
- Diagrame los Registros de activación o marco de pila de main y sumaComplejos en el punto A.

Considere int = 4 bytes, double = 8 bytes, puntero = 4 bytes, byte = 1 byte, char = 1 byte.

Pregunta 3 (2 puntos)

Considerando el código de la pregunta anterior, si el paso de parámetros se hiciera por referencia.

- Diagrame los Registros de activación o marco de pila de main y sumaComplejos en el punto A.
- El número de bytes que se almacenan en el registro de activación en esta forma de paso de parámetros ¿es mayor o menor que en el caso anterior?
- ¿Diría usted que es más eficiente? Explique indicando ventajas y desventajas.

Pregunta 4 (2 puntos)

Diagrame el registro de activación o marco de pila para la siguiente definición de función, señalando el desplazamiento de cada elemento con respecto al frame pointer.

```
int funcion1(char cc, int ii, char p[], double dd)
{
    char c;
    int a[4];
    struct dato {
        int uid;
        double peso;
    } sdato;
    ...
}
```

Pregunta 5 (6 puntos)

Para el siguiente código fuente en lenguaje C

```
int fibonacci(int n)
{
    int a = 0, b = 1, f = -1, i;
    if (n == 1)
        f = a;
    else
        if (n == 2)
            f = b;
```

```
        else {
            i = 3;
            while ( i <= n ) {
                f = a + b;
                a = b;
                b = f;
                i++;
            }
        }
        return f;
    }

int main()
{
    int n;
    scanf("%d", &n);
    printf("fibonacci(%d) = %d\n", n, fibonacci(n));
    return 0;
}
```

- a) Genere el código intermedio correspondiente en código de tres direcciones.
- b) Genere el código intermedio correspondiente en código P.
- c) Genere el código ensamblador para x86.