

SEGUNDA GUIA DE COMPILADORES

Falso y verdadero

- 1.-En lenguaje C los parámetros formales son como variables locales que ya fueron inicializadas en el momento de la llamada a la función o procedimiento
- 2.-En lenguaje C las variables locales (no estáticas) se crean cuando se entra a una función y se destruyen cuando se sale de la función
- 3.-En hoc los parámetros usados dentro de la definición de las funciones no tienen nombre
- 4.-No es posible definir funciones recursivas en hoc
- 5.-En hoc no hay variables locales
- 6.-En hoc cuando una función termina su ejecución se saca su marco de la pila de llamadas.
- 7.-En hoc los parámetros reales son listas de expresiones
- 8.-En hoc el código que ejecuta la máquina virtual de pila está en prefijo (considere como se ejecuta una operación de suma)
- 9.-En hoc el tipo de las variables es entero
- 10.-En hoc los parámetros reales se meten a la pila

Opción múltiple

- 1.-¿Qué es una macro en C?
 - a) un alias
 - b) algo que cuando el compilador (o preprocesador) encuentra lo sustituye con otra cosa
 - c) un nombre para un conjunto de instrucciones
- 2.-¿Qué es una macro con parámetros en C?
 - a) un alias
 - b) una función
 - c) una macro a la que se le dan los parámetros dentro de parentesis y separados por comas
 - d) un nombre para un conjunto de instrucciones
- 3.-En hoc2 ¿qué se usa como tabla de símbolos?
 - a) un arreglo
 - b) una pila
 - c) una lista simplemente ligada
 - d) una unión
- 4.-En hoc3 ¿qué se usa como tabla de símbolos?
 - a) un arreglo
 - b) una pila
 - c) una lista simplemente ligada
 - d) una unión
- 5.-¿Cuál es el tipo de las variables en hoc2?
 - a) int
 - b) float
 - c) double
 - d) union
- 6.-¿Cuál es el tipo de las variables en hoc3?
 - a) int
 - b) float
 - c) double
 - d) union
- 7.-¿Cuál es el tamaño máximo del nombre de las variables en hoc2?
 - a) 1
 - b) 2
 - c) 3
 - d) el tamaño del arreglo sbuf menos uno
- 8.-¿Cuál es el tamaño máximo del nombre de las variables en hoc3?

a) 1 b) 2 c) 3 d) el tamaño del arreglo subf menos uno

9.-En hoc2 donde se almacena el valor de una variable?

- a) en la posición del arreglo que corresponde a esa variable
- b) en la parte val de la unión de la entrada en la tabla de símbolos donde se guardó el nombre de la variable
- c) en la RAM de la Máquina Virtual de Pila
- d) en la Pila

10.-En hoc3 donde se almacena el valor de una variable?

- a) en la posición del arreglo que corresponde a esa variable
- b) en la parte val de la unión de la entrada en la tabla de símbolos donde se guardó el nombre de la variable
- c) en la RAM de la Máquina Virtual de Pila
- d) en la Pila

11.-Que es un apuntador?

- a) una dirección de una localidad de memoria
- b) una variable que almacena la dirección de una localidad de memoria
- c) una variable

12.-Para que sirve un apuntador?

- a) para almacenar la dirección de una localidad de memoria
- b) para manejar las cosas de forma indirecta
- c) para paso de parámetros por referencia

13.-Que introduce un apuntador simple?

- a) un nivel de indirección
- b) dos niveles de indirección
- c) tres niveles de indirección
- d) cuatro niveles de indirección

14.-Que introduce un apuntador doble?

- a) un nivel de indirección
- b) dos niveles de indirección
- c) tres niveles de indirección
- d) cuatro niveles de indirección

15.-Que introduce un apuntador triple?

- a) un nivel de indirección
- b) dos niveles de indirección
- c) tres niveles de indirección
- d) cuatro niveles de indirección

16.-Por que los apuntadores hacen muy poderoso a lenguaje C?

- a) hacen los programas de C mas fáciles de entender
- b) permiten manejar las cosas de forma flexible
- c) permiten que haya apuntadores salvajes

17.-Que valor proporciona el nombre de un arreglo?

- a) la dirección del arreglo
- b) la dirección del primer elemento del arreglo
- c) la dirección del último elemento del arreglo

18.-Que valor proporciona el nombre de una función?

- a) la dirección de la función
- b) la dirección de la primera instrucción de la función

c) la direccion de la ultima instruccion de la funcion

19.-Que es un apuntador a funcion?

- a) una variable que almacena la direccion de una funcion
- b) una variable que almacena la direccion de la primera instruccion de una funcion
- c) una variable que almacena una funcion

20.-Para que sirve un apuntador a funcion?

- a) para almacenar la direccion de la primera instruccion de una funcion
- b) para llamar a una funcion
- c) para llamar de forma indirecta a una funcion

21.-Que es double (*a)(double)?

- a) un double
- b) una apuntador a un double
- c) una funcion cuyo tipo de retorno es double y que recibe como parametro un double
- d) una funcion cuyo tipo de retorno es apuntador a double y que recibe como parametro un double
- e) una apuntador a funcion cuyo tipo de retorno es double y que recibe como parametro un double

22.-Que hace la funcion lookup?

- a) crea una entrada de la tabla de simbolos, la llena a partir de los parametros y la inserta al inicio
- b) crea una entrada de la tabla de simbolos, la llena a partir de los parametros y la inserta al final
- c) verifica si la cadena que recibe como parametro es el nombre de una entrada de la tabla de simbolos
- d) guarda las constantes y los builtins en la tabla de simbolos

23.-Que hace la funcion install?

- a) crea una entrada de la tabla de simbolos, la llena a partir de los parametros y la inserta al inicio
- b) crea una entrada de la tabla de simbolos, la llena a partir de los parametros y la inserta al final
- c) verifica si la cadena que recibe como parametro es el nombre de una entrada de la tabla de simbolos
- d) guarda las constantes y los builtins en la tabla de simbolos

24.-Que hace la funcion init?

- a) crea una entrada de la tabla de simbolos, la llena a partir de los parametros y la inserta al inicio
- b) crea una entrada de la tabla de simbolos, la llena a partir de los parametros y la inserta al final
- c) verifica si la cadena que recibe como parametro es el nombre de una entrada de la tabla de simbolos
- d) guarda las constantes y los builtins en la tabla de simbolos

25.-Cuantas pilas hay en hoc1?

- a) 1 b) 2 c) 3 d) ninguna

26.-Cuantas pilas hay en hoc4?

a) 1 b) 2 c) 3 d) ninguna

27.-Cual es el tipo por default de los elementos en la pila de YACC?

a) int b) float c) double c) union

28.-Cual es el tipo de los elementos en la pila de YACC en HOC1?

a) int b) float c) double c) union

29.-Cual es el tipo de los elementos en la pila de YACC en HOC2?

a) int b) float c) double c) union

30.-Cual es el tipo de los elementos en la pila de YACC en HOC3?

a) int b) float c) double c) union

31.-Cual es el tipo de los elementos en la pila de YACC en HOC4?

a) int b) float c) double c) union

32.-Que es void (*p)(double)?

a) una apuntador a void

b) una funcion cuyo tipo de retorno es void y que no recibe parametros

c) una funcion cuyo tipo de retorno es apuntador a void y que no recibe parametros

d) una apuntador a funcion cuyo tipo de retorno es void y que no recibe parametros

33.-en la funcion constpush se usan?

a) la pila de yacc, la tabla de simbolos y la RAM de la Maquina Virtual de Pila (MVP)

b) la pila de la MVP, la tabla de simbolos y la RAM de la MVP

c) la pila de la MVP y la tabla de simbolos

d) la tabla de simbolos y la RAM de la MVP

34.-Que hace la funcion initcode?

a) almacena el primer elemento del arreglo stack en stackp y el 1er elemento del arreglo prog en progp

b) almacena la direccion del primer elemento del arreglo stack en stackp y almacena la direccion del 1er elemento del arreglo prog en progp

c) guarda las constantes y los builtins en la tabla de simbolos

d) ejecuta una serie de instrucciones iniciado en la localidad de la RAM de la MVP cuya direccion se le pasa como parametro y termina cuando llega a un STOP

35.-Que hace la funcion code?

a) almacena la instruccion que se le pasa como parametro en la RAM de la MVP

b) almacena la instruccion que se le pasa como parametro en la la tabla de simbolos

c) almacena la instruccion que se le pasa como parametro en la pila de la MVP

d) ejecuta la instruccion que se le pasa como parametro

36.-Que valor retorna la funcion code?

a) la direccion de la RAM de la MVP donde se almacenara la siguiente instruccion

- b) una direccion de la RAM de la MVP
- c) la direccion de la RAM de la MVP donde se guardo la instruccion que se le pasa como parametro
- d) un apuntador a Inst

37.-En la funcion add se hace lo sig?

- a) se toman 2 operandos de la RAM de la MVP, luego se suman y la suma se guarda en la RAM de la MVP
- b) se toman 2 operandos de la RAM de la MVP, luego se suman y la suma se mete a la pila de la MVP
- c) se sacan 2 operandos de la pila de la MVP, luego se suman y la suma se mete a la pila de la MVP
- d) se sacan 2 operandos de la pila de la MVP, luego se suman y la suma se mete a la tabla de simbolos

38.-Como genera hoc4 el codigo?

- a) como si estuviera en notacion en notacion prefija
- b) como si estuviera en notacion en notacion infifija
- c) como si estuviera en notacion en notacion postfija

39.-Que hace la funcion execute?

- a) ejecuta una serie de instrucciones iniciado en la localidad de la RAM de la MVP cuya direccion se le pasa como parametro y termina cuando llega a un STOP
- b) ejecuta una serie de instrucciones iniciado en la primera localidad de la RAM de la MVP y termina cuando llega a un STOP
- c) almacena la instruccion que se le pasa como parametro en la RAM de la MVP
- d) almacena la instruccion que se le pasa como parametro en la pila de la MVP

40.-Cual es el tipo de los elementos en la pila de la MVP?

- a) int b) double c) Symbol d) Datum e) Inst

41.-Cual es el tipo de los elementos en la RAM de la MVP?

- a) int b) double c) Symbol d) Datum e) Inst

42.-Que almacena pc?

- a) almacena la direccion de la localidad de memoria actual
- b) almacena la direccion de la localidad de memoria donde esta la siguiente instrucción a ejecutarse
- c) almacena la direccion de la localidad de memoria de la RAM de la MVP donde se almacenara la siguiente instruccion
- d) almacena la direccion de la localidad de memoria donde esta la direccion de la 1er instruccion del cuerpo de una funcion o un procedimiento

43.-que almacena progp?

- a) almacena la direccion de la localidad de memoria actual
- b) almacena la direccion de la localidad de memoria donde esta la siguiente instrucción a ejecutarse
- c) almacena la direccion de la localidad de memoria de la RAM de la MVP donde se almacenara la siguiente instruccion

d) almacena la direccion de la localidad de memoria donde esta la direccion de la 1er instruccion del cuerpo de una funcion o un procedimiento

44.-Que funcion genera el codigo en hoc4

a) yylex() b) yyparse() c) code() d) execute() e) initcode()

45.-Si la condicion del while se cumple para que el cuerpo del while se repita se debe hacer?

b) guardar en pc la direccion de la primera instruccion de la condicion
b) guardar en pc la direccion de la primera instruccion del cuerpo del while
c) guardar en pc la direccion de retorno

46.-que almacena progbase?

a) almacena la direccion de la localidad de memoria actual
b) almacena la direccion de la localidad de memoria donde esta la siguiente instrucción a ejecutarse
c) almacena la direccion de la localidad de memoria de la RAM de la MVP donde se almacenara la siguiente instruccion
d) almacena la direccion de la localidad de memoria donde esta la direccion de la 1er instruccion del cuerpo de una funcion o un procedimiento definido por el usuario

47.- Que se almacena en un marco de funcion en lenguaje C

a) el nombre de la funcion llamada, los parametros
b) los parametros, direccion de retorno, las variables locales
c) los parametros, el contador de programa, las variables locales

48.-Que se almacena en la pila de llamadas

a) simbolos gramaticales
c) resultados de evaluacion de expresiones
b) marcos de funcion
d) tokens

49.-Que se almacena en el miembro ptr de la union u?

a) la direccion de la primera instruccion de una funcion builtin
b) la direccion de la ultima instruccion de una funcion builtin
c) la direccion de la primera instruccion de una funcion definida por el usuario
d) la direccion de la ultima instruccion de una funcion definida por el usuario
e) la direccion del primer elemento de un arreglo

50.-Que se almacena en el miembro defn de la union u?

a) la direccion de la primera instruccion de una funcion builtin
b) la direccion de la ultima instruccion de una funcion builtin
c) la direccion de la primera instruccion de una funcion definida por el usuario
d) la direccion de la ultima instruccion de una funcion definida por el usuario
e) la direccion del primer elemento de un arreglo

51.-De lo mencionado abajo donde no hay saltos del contador de programa

a) secuencia
c) ciclos
b) decisiones (if)
d) funciones y/o procedimientos

52.-Que almacena stackp?

- a) almacena el tope de la pila de YACC
- b) almacena el tope de la pila de la Maquina Virtual de Pila (MVP)
- c) almacena la direccion de la localidad de memoria donde esta la siguiente instrucion a ejecutarse
- d) almacena la direccion de la localidad de memoria de la RAM de la MVP donde se almacenara la siguiente instrucion
- e) almacena el tope de la pila de llamadas

53.-Que almacena fp?

- a) almacena el tope de la pila de YACC
- b) almacena el tope de la pila de la Maquina Virtual de Pila (MVP)
- c) almacena la direccion de la localidad de memoria donde esta la siguiente instrucion a ejecutarse
- d) almacena la direccion de la localidad de memoria de la RAM de la MVP donde se almacenara la siguiente instrucion
- e) almacena el tope de la pila de llamadas

54.-Que ejecuta execute(sp->u.defn)

- a) el cuerpo de un builtin
- b) el cuerpo de un ciclo
- c) el cuerpo de una funcion definida por el usuario
- d) la primera instrucion de una funcion definida por el usuario

55.-De lo de abajo que no hace la funcion ret

- a) saca los parametros de la funcion de la pila
- b) saca el marco de funcion al que apunta fp de la pila de llamadas
- c) guarda en pc la direccion de retorno
- d) llama a procret
- e) hace que fp apunte al inicio de la pila de llamadas

Que codigo genera hoc si el usuario escribe lo de abajo

- a) 100
- b) 33 + 45
- c) 5 + 9 * 2
- d) x+y
- e) (2+x)*(y+5)
- f) aa = exp (2 (
- g) aa = 10 * sin(PI/4)
- h) bb + cc * (dd+ee*100)
- i) i = 0 while (i < 10) {
 print i
 i =i + 1
}
- j) x=suma(300, 400)
- k) y=suma(1+3, x-5)