The four-byte sequence 0xD6 0x85 0x41 0xA0 stored in consecutive memory cells in a little-endian architecture represents _____ (decimal) when interpreted as a 32-bit signed integer.

-1606318634

The four-byte sequence 0x8B 0xB1 0x7C 0x96 stored in consecutive memory cells in a little-endian architecture represents _____ (decimal) when interpreted as a 32-bit signed integer.

-1770212981

The four-byte sequence 0xF6 0x45 0x71 0x85 stored in consecutive memory cells in a little-endian architecture represents _____ (decimal) when interpreted as a 32-bit signed integer.

-2056174090

The four-byte sequence 0xC8 0x85 0xFF 0x21 stored in consecutive memory cells in a little-endian architecture represents _____ (decimal) when interpreted as a 32-bit signed integer.

570394056

The four-byte sequence 0x22 0x86 0x82 0x56 stored in consecutive memory cells in a little-endian architecture represents _____ (decimal) when interpreted as a 32-bit unsigned integer.

1451394594

Values passed to a subroutine by a calling program are called _____.

arguments

What general types of parameters are passed on the stack?

- [ ] Parent-child arguments
- [ ] Evaluation arguments
- [ ] Legal arguments
- [ ] Context-free arguments
- [x] Reference arguments
- [x] Value arguments

Local variables are created by adding a positive value to the stack pointer.

- ( ) True
- (•) False

What advantages do stack parameters have over register parameters?

- [x] Stack parameters reduce code clutter because registers do not have to be saved and restored.
- [ ] Register parameters are optimized for speed.
- [ ] Programs using stack parameters execute more quickly.
- [x] Stack parameters are compatible with high-level languages.

Which offers a more flexible approach, passing arguments to procedures in registers, or on the stack?

- ( ) in registers
- (•) on the stack

A stack frame is _____

- ( ) An area in the heap that is used to store global variables
- ( ) The area of the stack set aside for storing global strings.
- (•) The area of the stack set aside for passed arguments, subroutine return address, local variables, and saved registers.
- ( ) A register window pointing to local variables.
- ( ) The area of the text segment set aside for passed arguments, subroutine return address, local variables, and saved registers

Which of the following shows the procedure entry code generated by MASM when the LOCAL directive is used to declare a doubleword variable?

- ( )
```
mov ebp,esp
push ebp
add esp,4
```
- (•)
```
push ebp
mov ebp,esp
sub esp,4
```
- ( )
```
push ebp
mov esp,ebp
add esp,4
```

```
    mov ebp,esp
push ebp
sub esp,4
```

High-level languages always pass arrays to subroutines by value.

- True
- False

A subroutine's stack frame always contains the caller's return address and the subroutine's local variables.

- True
- False

Passing by reference requires popping a parameter's offset from the stack inside the called procedure.

- True
- False

Which of the following defines an array local variable consisting of 50 signed words?

- LOCAL SWORD:wArray[50]
- LOCAL wArray[50]:SWORD
- LOCAL SWORD[50]:wArray
- LOCAL wArray:SWORD[50]

Place the stesp for creating a stack frame in the correct order

1

Passed arguments, if any, are pushed on the stack.

2

The subroutine is called, causing the subroutine return address to be pushed on the stack.

3

As the subroutine begins to execute, EBP is pushed on the stack.

4

EBP is set equal to ESP. From this point on, EBP acts as a base reference for all of the subroutine parameters.

5

If there are local variables, ESP is decremented to reserve space for the variables on the stack.

6

If any registers need to be saved, they are pushed on the stack.

What are the two common types of stack parameters?

- [x] Value parameters
- [ ] Static parameters.
- [ ] Object parameters.
- [x] Reference parameters
- [ ] Formal parameters.
- [ ] Abstract parameters.

Arrays are passed by reference to avoid copying them onto the stack.

- (•) True
- ( ) False

Another name for a stack frame is

- (•) Aviation record
- ( ) Vinyl record
- ( ) Stack record
- ( ) Local storage
- ( ) Heap record

- Activation record

An argument passed by reference consists of the offset of an object.

- True
- False

When an argument is passed by value, a copy of the address is pushed on the stack.

- True
- False